MINIMUM DISTORTION DATA HIDING FOR COMPRESSED IMAGES

A Thesis
Presented to
The Academic Faculty

by

Çağatay Candan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
January 2004

MINIMUM DISTORTION DATA HIDING FOR COMPRESSED IMAGES

Approved by:

Dr. Nikil Jayant, Adviser

Dr. John Barry

Dr. Russell Mersereau

Dr. Elizabeth Mynatt
(College of Computing)

Dr. Monson Hayes

Date Approved: 09 January 2004

ACKNOWLEDGEMENTS

Many people have contributed to this work. It is impossible to name all of them. I am thankful to everyone who has been with me during this incredible journey.

I would like to thank my advisor Nikil Jayant first. He has been a role model to all of us. He has always provided support whenever we need, and never stopped believing in what we can do. His brilliance, attention and positive attitude towards life is always to be remembered. While working with Dr. Jayant, I had the chance of observing not only how a brilliant person tackles research problems, but how a great director interacts with people from very difficult backgrounds and walks of life to develop something together.

I would like to thank Dr. Mersereau and Dr. Hayes next. I would like to thank them for the courses they have prepared for us and for their great contributions to CSIP and GCATT and to the academic culture of the school. Nothing would be the same without them. I would also like to thank Dr. Mynatt and Dr. Barry for being in my thesis committee.

Finally, I would like to thank my friends, my supporters during difficult times and sources of joy in good ones. I believe Georgia Tech is great because of its students; not the other way around. I will always remember the days I was surrounded with curious, smart people; discussing books, movies and life in general. For some reason I am thinking that our days were the best days of my life. I hope life after school would prove me wrong. Here is a list that I want to keep within this document: Brian, Roberto, Babak, Wajih, Jang-Hyun (Yoon), Renato, Mai, Apu, Estuardo, Elizabeth, Sermet, Borte, Erdem, Deniz. Take good care, do good work and keep in touch :)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

We present a novel data hiding method for compressed images. The method is designed to minimize the quality loss associated with data embedding into a JPEG image. The described technique uses the objective criterion such as the mean square error and the human visual system based criterion such as the Just Noticeable Distortion metric for the distortion minimization. The hiding method is designed under the restrictions of the JPEG compression standard to develop new image applications without any modifications or additions to the existing standard. An application example is presented in the thesis. The performance of the technique is examined at different image sizes and resolutions. The cost of hiding in terms of file length extension is examined. Some subjective experiments to determine the zero perceived distortion hiding capacity are made. An application illustrating the usage of the technique is given. The described application embeds check-bits into JPEG images to facilitate the verification of the sender identity and the authenticity of the transmitted image. In this thesis, we give a list of requirements on the data hiding methods to implement standard compliant applications; design a provably good hiding method operating under these requirements; determine the critical performance points of the method and propose an application based on the method.

We have performed some additional research to determine how our system works with high resolution images and existing other well-known algorithms for information hiding. The experiments on the high-resolution images have shown that there exists a large embedding capacity for the high-resolution images in spite of a loss of embedding density. The performance comparison experiments have shown that the spread spectrum technique offers a competitive but less efficient distortion performance.

CHAPTER I

INTRODUCTION

We present a novel data hiding method for JPEG compressed images. The method minimizes the distortion associated with the embedding-compression operation. The goal of the thesis is to minimize the quality loss, or to maximize the quality of the images after compression and hiding. An application to the method is proposed to illustrate its usage.

The described data embedding method is designed for the content based image communication applications. The content based applications that are realized with this method do not require any modifications or additions to the existing compression standard. The incurred cost of the content based applications is the additional distortion on the transmitted due to hiding. The perceptibility of the introduced distortion is minimized in this thesis. The described algorithm can be used as follows: The application data is embedded in a minimally distortive way into the carrier. The carrier propagates in the conventional image communication channel which can be the cable tv system, or an IP based video-on-demand system. Upon the arrival of the carrier to the receiver, the display image is decoded with the conventional image decoders. If the customer is a subscriber of the embedded application, the data is extracted from the image and forwarded to the application unit. The project presented here is strictly limited to still images, its extension to video is one of our future works.

In this thesis, we give a list of requirements on data hiding to implement content based image communication applications; design a provably good method under these restrictions; determine the performance limits of the method and design an application for the method.

In this chapter we present a brief description of data hiding and examine the milestones of our project. The project is presented with an application example on video broadcasting. The video application is described to illustrate our motivation, the requirements of the communication applications and to emphasize the application range for video signals.

Information Hiding

Information hiding is the addition of an application oriented information to a multimedia signal without causing any perceptible distortion. The energy of the embedded signal should be low enough when projected onto the human perception domain, but it should be strong enough for a robust machine detection.

The information hiding applications can be classified into two categories. The embedded data of some applications such as image captioning, feature embedding etc. can be desirable for all parties involved with the signal. In some other applications, such as the copyright protection, a party may benefit from the existence or non-existence of the hidden information. The difference in the application domain shapes the requirements and features of the data hiding algorithms.

The features of the information hiding methods can be listed as follows:

Blindness: In some applications, the cover signal can be used at the decoder to extract the hidden information. The system is said to operate blindly if it is operating in the absence of the original signal.

Robustness: The robustness to attacks, which can be accidental or intentional, can be critical in some applications.

Cryptographic Security: The cryptographic security of the hiding method for the copyright and related applications needs to be established.

Capacity: The amount of information that can be embedded at a distortion

tolerance is the payload of the hiding algorithm. The hiding capacity of an image is the maximum of the achievable rates.

Distortion: The quality of the after-hiding signal is important at all applications, but the distortion handling is especially critical for the applications demanding high quality signal replicas as in digital television systems, megapixel cameras etc.

The algorithm complexity, the key management protocols are some other issues related to the data hiding algorithm design.

A list of applications with their requirements is given in Table 1.

Table 1: Applications of Image Data Hiding and Their Requirements

Application	**Carrier Signal**	**Hidden Signal**	**Attacks**	**Capacity**
Watermarking	Known	Known	Deliberate	Low
Captioning	Unknown	Unknown	Accidental	Variable
Steganography	Known/Unknown	Unknown	Accidental	High
Error Protection	Known	Correlated	Accidental	Variable
Content Control	Unknown	Known	Deliberate	Low

Compression and Hiding

Images with hidden data suffer from two noise sources. The first source is the quantization noise due to the compression operation. The second one is the embedding noise due to hiding. Uncompressed images contain significant amount of redundancy which can be used for data hiding purposes. But practically most images other than the military, archival or legally sensitive ones are stored and transmitted in the compressed formats.

The compression operation removes the structural and the statistical redundancy from the source. The structural redundancy is removed at the the transform domain. The transform domain coding decorrelates or untangles information units of the image. The decorrelated information units are approximated finely or coarsely

or discarded totally depending on their contribution to the quality. The described quantization stage introduces the quality loss. The statistical redundancy of the quantized source is removed by assigning shorter labels for the popular words, i.e. lossless coding.

The measurement of image quality is critical for compression and hiding. The mean square error (MSE) and its weighted versions have been used extensively for this purpose. MSE is known to be far from ideal at distortion measurement. It can be tricked by introducing a structured noise to which the human eye is sensitive, but the metric is not. An example of one noise source is the salt-and-pepper noise whose effect is neglected by MSE because of its averaging based definition. MSE is known to work well when the distortion has no inherent structure (uniform noise). The performance of the JPEG and other quantization based systems are measured with MSE (equivalently PSNR) when there are no systematic artifacts such as blocky artifacts.

The components of the human visual system are analog devices with finite discretion capabilities. The saturation effect of the eye is one of the threshold effects that is not modeled by any mathematical quality metrics. The Figure 1 shows the rate-distortion curves for MSE and the perceived distortion metrics. As expected the perceived distortion decreases with the increased bitrate. But there exists a critical point on the perceived distortion curve that further increase of bitrate beyond $R_p(0)$ does not bring any improvement of perceived quality. On the other hand, the MSE metric is a mathematical norm and can not model such threshold effects. MSE reduces to zero only if the compressed image is identical to the original image, hence there is no loss.

The threshold effect can also be examined in the data hiding context. In Figure 2, the cameraman images carry different resolution versions of the tech tower image. The lowest resolution image causes no perceived distortion. As the resolution (the

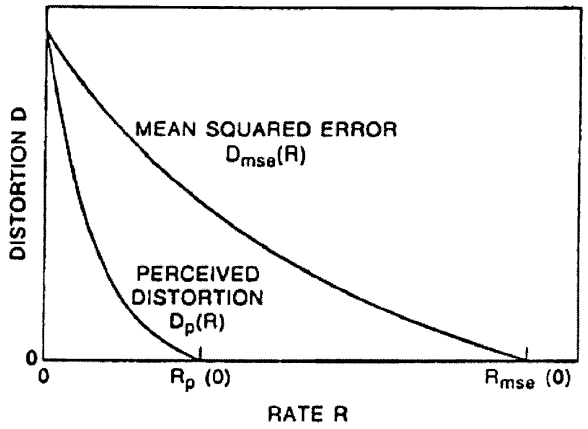


Figure 1: Rate-Distortion Curves for MSE and Perceived Distortion

number of DCT coefficients embedded) is doubled; the distortion becomes perceptible at some parts of the carrier image. When it is increased by eight folds, the embedding noise becomes excessive.

Thesis Statement

The goal of the thesis is to design a minimally distortive data hiding algorithm. The operation domain is the JPEG images. The designed algorithm should operate at any compression and hiding bitrate pairs. To sustain the compatibility with the JPEG standard, the algorithm should operate blindly or in other words should operate with a pre-determined, input independent set of rules. The decoder of the algorithm should operate at a low computational cost. The encoders are allowed to operate at a higher computational cost. The higher cost for the encoder operation is allowable since the number of the encoders in a typical communication loop is a lot fewer than the decoders.

The central challenge of the project is the distortion minimization. The requirement of JPEG domain operation lets us to give an analytical distortion characterization of the quantization error on DCT coefficients before and after hiding. The

Figure 2: Embedding Noise at Different Levels

results of this analysis is used to find the optimal ways of transmitting hidden bits in a statistical sense. The optimal ways of embedding and extracting bits form the pre-determined rules of hiding. To reduce the perceived distortion on a given image (not the ensemble), the data embedding density is varied from block to block. The data embedding density is determined in a such a way that the system blindness (input independent operation) is preserved. The resolution of the clash between the opposing requirements of the blindness and the minimum distortion embedding is the central challenge of the project.

Project Motivation

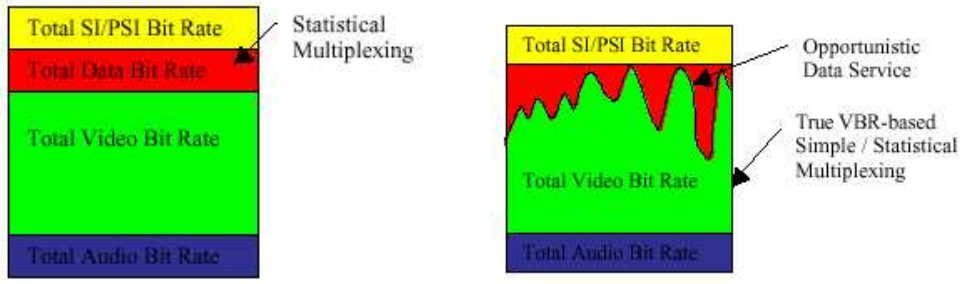
The motivation of the project is discussed with an application example. This discussion is aimed to show the practicality of the research and to emphasize the importance of the requirements already discussed. We illustrate a data broadcasting

application. The application data is embedded and broadcasted to the users in the video signal. One may think of the system as a cable television-internet system with data hiding.

The standard broadcast systems use separate channels for video and data broadcasting. The data services are the additional features that the customers may choose to subscribe. The main of purpose of the broadcast systems is to provide the video signal. Therefore the secondary data services use the leftover bandwidth that is not consumed by the video signal. In order to have more data applications, the cable operators compress the video signal as much as it is possible. One way of data bandwidth assignment is to use fix bandwidth per video channel and use the remaining constant bandwidth for the data applications. Another way is to use variable bitrate coding for video channels and use the time varying leftover bandwidth for data applications. The variable bit rate coding uses less bits than the average number of bits at low activity scenes, therefore saves more bits for data applications. The only reason of the usage of the complicated variable bitrate coding over the constant bitrate coding is the data applications. The variable bit rate coding can not be used to increase the number of video channels offered. The number of channels is calculated at the worst case conditions of having high activity scenes at all channels. You can see an illustration of two coding system in Figure 3. The VBR coding allocates more bandwidth for the data services, but it should be noted that these services can not be used for the real-time applications, since the VBR is an opportunistic system incapable of providing a dedicated service, [27].

Data hiding can present an alternative to the VBR coding for data broadcasting. Instead of compressing the video signal to provide bandwidth to data applications, we may prefer to embed the application data directly into the video signal.

The data broadcasting with hiding has two costs. The first one is the increment in bandwidth, if any, due to the embedding operation, i.e. bandwidth cost. The second



(a) Fixed Bitrate Allocation (b) Variable Bitrate Allocation

Figure 3: Fixed and Variable Bit Rate Video Broadcasting, [27]

one is the distortion on the picture. If the length of the video signal before and after data hiding does not change (or marginally change), the overall cost burden of data applications is on the picture distortion. Our goal in this thesis is to minimize this cost. Hiding at the cost of zero perceived distortion is particularly interesting to us.

We show a block diagram of a typical data broadcasting system with hiding in Figure 4. The application data is embedded into the carrier signal at the transmitter. The carrier signal propagates through the conventional communication channel and reaches the destination point. At the destination the video signal is decoded and displayed. And the hidden data is extracted and forwarded to the application.

We list the advantages and the disadvantages of the proposed system. The advantages are:

- Synchronization

  The content and its data are delivered synchronously. There is no latency problem, the data transmission protocols can be simplified.

- Application Programming

  Some applications use reference points in the video signal (like add placement, pop-up video). Data embedding can be a more convenient programming option for these applications.

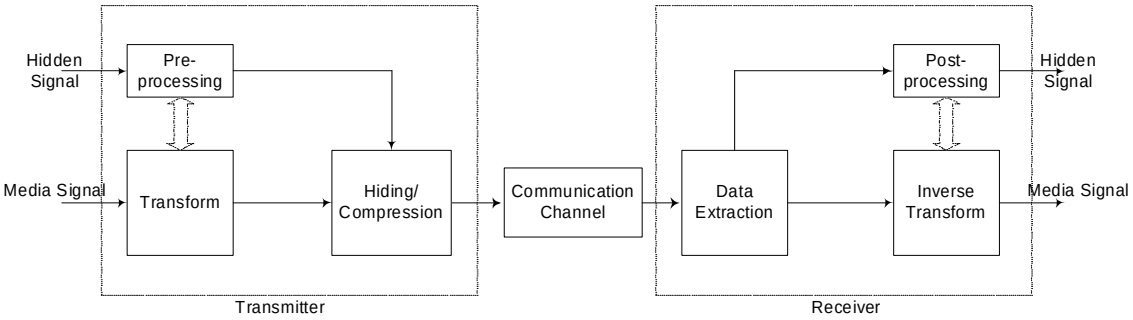


Figure 4: Block Diagram for Compression-Hiding System

- Permanent Data Bandwidth

  The image quality is a continuous function of bitrate (i.e. there are no sudden jumps of quality, when the bitrate is changed a little). There may exist a data embedding rate where it is possible to have a permanent data channel between the server and the customer's receiver without any perceptible quality degradation or bandwidth increase.

The disadvantages are:

- Distortion at high embedding rates

  The accumulated distortion can be visible at high embedding rates. It should be noted that the embedded data always acts as a noise source second to the compression noise.

- Better VBR coding eliminates the hiding alternative

  The embedding option uses the inefficiencies of the VBR coding. As more efficient coding methods are developed and adopted as standards, the utility of the hiding alternative vanishes.

The goal of the thesis is to design an algorithm which can be used at data communication applications. The requirements on the algorithm can be listed as:

- Minimum Distortion

- Blind Decoding
- JPEG compatibility
- Efficient implementation
- Scalability in both compression and embedding bitrates.

Research Summary

The distortion minimization is the main goal of the project. To achieve the distortion minimization goal, we propose a novel approach of densely partitioning the JPEG codewords (JPEG quantization words) into different classes. Each class represents a hidden bit combination. The hidden data encoder searches the elements of the class determined by a given set of hidden bits to find the closest word to the original. The decoder extracts the label of the class (i.e. hidden information) with a simple parity check.

The central issue of the research is to determine good partitioning strategies to partition the JPEG words into classes. A good partitioning method should be scalable (variable embedding rate), i.e. it should partition the set into an arbitrary number of subsets. It should also retain the good approximation properties of JPEG compression at every partition. To find the best partitioning strategy, we analyze the expected distortion per hidden bit. Using the results of this analysis, we have executed an exhaustive search to find the best partitioning strategies at different compression-embedding bitrate pairs. The analysis is based on the quantization noise on DCT coefficients. We examine the combined effect of compression and hiding to find the expected perturbation of DCT coefficients at the final output. Running a similar exhaustive search for the optimal partitions at every possible embedding and compression rate is an impossible task. We have proposed a simple yet efficient adhoc

partitioning strategy (recipe) which tracks the optimal strategies very closely at the known optimal compression/embedding rate pairs. The search for optimal partitioning strategies concludes the objective distortion minimization stage.

At the next stage of the distortion minimization, the local features of the carrier image are determined and used to diffuse the embedding distortion into the least visible regions of the carrier. To accomplish that, we determine the noise perceptibility thresholds of DCT coefficients (Just Noticeable Distortion levels) through the Watson's human visual system model [50]. JND information allows us to estimate the noise sensitivity level of different image regions such as smooth, busy, edge regions. MSE criteria is weighted by JND values and then the bits are embedded to minimize weighted MSE measure. With this process, the image regions with low noise sensitivity such as the low contrast or busy regions receive more than average number of hidden bits and the other regions with high noise sensitivity carry fewer hidden bits. Our goal at the second stage of the distortion minimization is to incorporate some image dependent subjective criterion to reduce the level of perceived distortion. According to the experimental results, hiding with subjective optimization (JND based approach) is superior at all embedding levels.

After the description of the method, we present the results of the subjective image quality tests to determine the zero perceived-distortion hiding capacity. The hiding capacity bitrates at the zero perceived distortion is the range of costless data transmission. We examine the zero perceived distortion capacity at different compression levels. The results of the experiment on the file length extension due to hiding is discussed. Some applications of the algorithm on high resolution images is given.

An application for the described data hiding method is presented next. The application is on JPEG image authentication. The sender of the JPEG images embeds an identification tag with some authentication bits derived from the content into the transmitted image. The receiver can check the integrity of the received image and

the identity of the sender through the embedded data.

Thesis Roadmap

The introduction chapter is intended as an extended project description with project motivation, requirements and research milestones. The areas of signal compression, information theory and information hiding is discussed in detail in the following chapter. The third chapter is the description of the designed method with the experimental results. The fourth chapter is the authentication application on the described method. The thesis concludes with final remarks.

CHAPTER II

INFORMATION HIDING

Data hiding methods can be divided into two categories:

- Robust Methods
- Non-Robust Methods

The attack robustness is the main concern for the methods of the first category. The distortion cost to achieve the robustness is not addressed explicitly. The methods of second category address the distortion problem. In this chapter, we examine these categories along with the relatively recent information theoretical developments on the subject.

2.1 Robust Data Hiding

The robust data hiding methods are also known as watermarking methods. In this document, the data hiding term is used as a more general term encompassing all hiding methods which can be robust or not. The watermarking techniques have been studied to a greater depth than the non-robust techniques. These studies have been motivated by the emerging needs of the audio-video content providers which are looking for the means to prevent the illegal distribution of their properties. Unfortunately to our knowledge, the challenge could not resolved until now. Readers may visit "DVD Copy Control Association" (www.dcaa.org) and "Secure Digital Music Initiative" (www.sdmi.org) web-sites to get more information on these efforts.

The main difficulty that has evaded the watermarking solution is the multiplicity of the attacks. Most engineering problems involve a passive adversary such as random

noise, friction or gravity; but in the watermarking problem the adversary is a human being who can analyze the system and better the attack strategy by time. The battle-of-wits situation between the designer and the attacker for this problem puts the designer at a disadvantage because of the multiplicity of attacks that needs to accounted for. Watermarking researchers adopted some cryptograhical techniques to discourage the attackers to no avail.

Some successful data hiding methods compensating different attacks have been proposed. The main ones of these methods are listed below:

Cox's Spread Spectrum Method: The hidden information (a single bit embedded to justify ownership) is modulated by a sequence of randomly generated bits and the modulated bits are algebraically added to the original image in the DCT domain [18]. At the detection step, the inner product (matched filter) of the modulation vector and the received signal minus the original signal is evaluated and the result is thresholded. The method depends on the ability of the decoder to find the hidden information bins (due to match filter operation). If the attacker rotates an image by 5 degrees, the detector should undo this attack by a rotation of -5 degrees before the evaluation of the inner product. A very similar synchronization problem occurs in the communication context. If the synchronization can be achieved, the spread spectrum embedding method works very well under many attack conditions, [19].

Adaptive Methods: Different from the random embedding approach of the spread spectrum method, the adaptive methods adjusts the embedded signal to the local features of the original image. In [47] the local characteristics of an image is first determined such as edge, uniform (non-edge) with low/high intensity, moderately/very/extremely busy (high frequency terms). The noise sensitivity of these classes are estimated and the signal is embedded accordingly. In [40], the hidden signal is shaped to be masked by the original signal. The computer simulations show that the throughput of the hiding system can be improved with these techniques.

Invariance Methods: A novel approach to counter the expected attacks is to insert the hidden signal into an invariant domain of the attack. If we desire the invariancy to shifting, we can embed the hidden data to the amplitude of the its Fourier transform. The shift invariancy idea is generalized to joint invariancy under shift, rotation and scaling in [37]. Due to the digitized nature of images, the domain invariant to shift-scale-rotation operations turns out to be difficult to implement, [37].

2.2 Minimum Distortion Data Hiding

The minimum distortion data hiding techniques operate on different principles. The hidden bits are directly embedded into the image instead of modulating them with a longer sequence as in the robust methods. This process is usually done with a quantization based operation. The robustness of the minimally distortive methods can be increased by an application of error-correction coding to the hidden bits. That is, a desired number of redundant bits (equivalent of bit repetitions) are padded to the original information bits via error-correction code. The quantization based minimum distortion data hiding methods are listed below:

Pixel Domain Quantization: The least significant bit of every pixel is replaced with a single bit of hidden information. This technique is the earliest information hiding idea in the literature.

Transform Domain Quantization (Chen's Quantization Index Modulation): The algorithm of Dr. Chen [13] generalizes the scalar quantization in pixel domain to the vector quantization in transform domain. This generalization is especially interesting to us because of its theoretical foundation and because of its linkage to our method. The embedder of this algorithm aims to create a rich superstructure with many sub-structures whose sub-structure index conveys the hidden bits. For example, the encoder can have two different sub-structures. The first one can be the images with all even valued pixels. The second one can be the images with all odd

pixels. The embedder can hide a single bit by modifying a given image to one of these structures. The distance between two structures is a criteria of importance for this system. In the stated example above, the distance between two classes is the number of pixels of the image. Dr. Chen visualizes this process as a codebook selection problem depending on the hidden bit. The title of the method (Quantization Index Modulation) is suggested by the operation of the technique. Readers can also examine [14] for further details.

The codebooks of this method are designed to have a minimum distance of d (that is a codebook (the union of codewords) has no word of another codebook at a distance less than d). If the codebooks are separated by the distance d, the attacked words can be safely recovered if the attack displaces the codeword with a distance less than $\frac{d}{2}$ units. The recovery process is easy to see from the example given in the previous paragraph.

As Dr. Chen indicates the method is one of the few provably good embedding techniques in the literature, [14]. We would like review some results on this technique that we return at the latter discussions.

The design of the codebooks can be visualized as follows: The encoder creates a good codebook for the quantization of the input. The encoder partitions this codebook into sub-codebooks which are separated by the distance d. It is important to note that the sub-codebooks should be partitioned in such a way that they retain their good quantization properties. In other words, the quantization performance of the sub-codes should not be very different from each other. After the codebooks are partitioned, the sub-codebooks are made public. To transmit a signal with the desired hidden bits, the encoder chooses a specific sub-codebook determined by the hidden bits. Then the encoder selects a codeword which is jointly typical with the input in that particular sub-codebook. The decoder examines the codewords of all codebooks and finds the codeword which is jointly typical with the received signal. The jointly

typical codeword is the decoded output and the index of the codebook in which the codeword resides is the hidden information, [13, 16, 15, 35]. We would like to note that our data hiding method described in the next chapter operates on very similar principles. Instead of finding the universal codebook, we use the JPEG codebook. And we partition the codebook into many sub-codebooks exactly as described. The details of the partitioning operation is given at the next chapter.

2.3 Hiding Capacity

The fundamental problem of communication is the transmission of information in a reliable way. The maximum of the achievable rates for the reliable communication is the capacity of the channel. The information hiding capacity is studied in the literature, [13, 35, 15] and the most significant contributions are due to Dr. Moulin, [35].

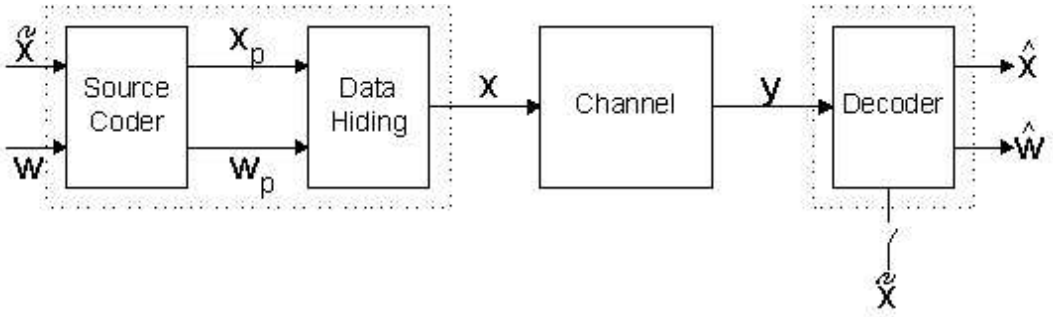


Figure 5: Data hiding system model

In the Figure 5 the components of a data hiding system is shown. The marked message propagates through an attack channel and reaches the decoder. The channel block is the attack block. Attacks can be accidental, like additive noise; or intentional, like the JPEG compression. In the conventional communication problems the attack phenomena is modeled as an additive noise which can be due to a malfunctioning device or an uncontrolled factor.

The breakthrough in theory has been accomplished by the recognition of an existing analogy [13, 35, 15]. The analogy models the attack information and the original signal as the state of the communication channel. The attackers can select the channel mode and the hidden data decoders operate on the noisy information on the channel state.

This analogy allows us to use some known results of information theory in the data hiding context. Pinsker analyzed the capacity of the channels with multiple states whose state information is not available to both parties in [24]. The Pinsker result establishes the basic foundation of the data hiding theory. An instructive illustration of Pinkser's theory is given by El Gamal for another application, [25]. El Gamal studies the capacity of the defective memory chips. The defective memory is modeled with three states. The states are "no permanent damage", "stuck-to-1" and "stuck-to-0" states (see Figure 6). The stuck-to states represent the values that are stored to the memory irrespective of the input. No permanent damage state can have occasional errors with ϵ probability.

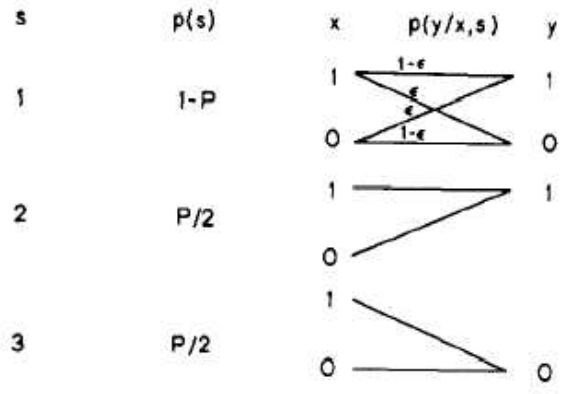


Figure 6: Memory with Defects

The state of the memory corresponds to the attack type from hiding view point.

The transmitted information (hidden info) is the bits that are written to the memory. The no-damage-state corresponds to the operation at which the attacker randomly flips the bits. The stuck-states correspond to the attacker overwriting the hidden information. The capacity analysis of the defective memory is reported by El Gamal in [25]. The results of Table 2 illustrates these results.

Table 2: The capacity values of the defective memory channel. The availability of the defect information at the encoder and the decoder is shown by the crosses and the checkmarks.

Encoder	Decoder	Capacity
$\times$	$\times$	$C_{min} = \max_{p(x)} I(X;Y)$
$\surd$	$\surd$	$C_{max} = \max_{p(x\|s)} I(X;Y\|S)$
$\surd$	$\times$	$C_{enc} = \max_{p(u,x\|s)}(I(U;Y) - I(U;S))$
$\times$	$\surd$	$C_{dec} = \max_{p(s_o\|s)} \max_{p(x\|s_o)} I(X;Y\|S)$

The Table 2 covers all possible modes of communication with channel state information. The state information not available to both encoder and decoder is the first mode. The availability of the state to both encoder and decoder is the second mode (data hiding with known cover signal and attack). The availability of the state only to the encoder is the third mode (hiding with unknown cover signal at the decoder, blind data hiding). The availability of the channel state only to decoder is the fourth mode (perfect attack recognition). We would like to examine the data hiding capacity results in the light of the defective memory example:

Non-Blind Information Hiding Systems: When the state information (the carrier signal and the attack type) is available at the encoding and the decoding sides, the system is called to be non-blind. Non-blind systems are simpler to analyze and easier to design. It is always a possibility to eliminate the effect of the carrier at the detector for non-blind systems.

The first signifacnt work on the non-blind systems has been given by Dr. Servetto [43]. At this analysis the pixels/transform coefficients are thought as the bins of

information storage. The embedder inserts the hidden data into the bins, while keeping the embedding distortion bounded. Similarly the attacker applies a bounded power distortion in order not to over distort the carrier.

The hidden bit k (W_k) is modeled to be attacked by an additive noise source. The power of the attack signal is limited by the attack variance bound. This leads to a modeling of $\tilde{W}_k = W_k + J_k$.

The attack on the information bins are correlated with the carrier. Even though the operation is non-blind, the carrier signal may leak and can not be totally cancelled out at the decoder for many attacks (irreversible attacks). But with the bounded power additive noise assumption, the best strategy for the attack is known to be independent of the input. A similar problem has been discussed in the communication with the intentional jamming scenario in the book of Cover and Thomas, [17]. Dr. Servetto refers to this problem for the solution of the hiding capacity.

The analysis of the hiding game is surprisingly simple. A payoff function for the information hiding game is defined: $J(W, J) = I(W; W + J)$. $I(W; W + J)$ is the mutual information of the input and output of the attack channel. The embedder and the attacker controls the distributions of W and J respectively. The aim of the embedder is to maximize the payoff function and the aim of the attacker is the opposite.

The game theory has a solution to this conflict of interest. At some games (not all), it is possible to define a saddle point condition which is optimal for both parties. The saddle point condition can be stated as:

$$J(W; J^*) \leq J(W^*, J^*) \leq J(W^*, J) \qquad \forall\, W, \forall\, J$$

The starred strategies in the above condition are the optimum strategies for two parties. The saddle point condition states that any deviation from the starred strategies is not beneficial to anyone. For every possible way of deviation, there exists a better counter strategy reducing the return below the return of starred strategy. It should

be noted that not all games have saddle point solutions.

For the information hiding game it is relatively straightforward to show that the saddle point condition exists. The optimum embedding and attack strategies turn out to be the normal distributions with zero mean and the variance matching the power limitations of the embedder and the attacker.

Blind Information Hiding Systems: Blind information hiding schemes are analyzed with the channel state described, [35]. The previously stated Pinsker paper and the El Gamal results are instrumental for the development of the theory.

The analysis of the blind hiding systems is similar to the Gaussian game of Servetto. The main difference is the modification of the payoff function $J(\tilde{Q}, Q) = I(U;Y) - I(U;\tilde{X})$. In here, $\tilde{X}$ denotes the cover signal. The embedder chooses the distribution $\tilde{Q}$ which maps $\tilde{X} \rightarrow (U, X)$. U is an auxiliary variable and X is the composite signal.

The attacker chooses the distribution Q which maps $X \rightarrow Y$. It is important to note that both the embedder and the attacker have a maximum distortion constraint between the input and output of their block. Therefore mapping $\tilde{Q}$ and Q can not be arbitrary, but has to include the quality considerations.

The capacity of the blind hiding system is determined by the following max-min relation. $C = \max_{\tilde{Q}} \min_{Q} J(\tilde{Q}, Q)$. The capacity value guarantees that if the rate of the hidden signal is below C, there exists a data hiding code with an arbitrarily small probability of wrong decoding under the given embedding and attack power constraints.

There are very few cases of the above max-min operation that can be evaluated, [35]. A surprising result of this theory is the equality of the capacity for blind and non-blind hiding schemes for the Gaussian distributions. This counter intuitive information theoretical result appears at another context in [16].

CHAPTER III

MINIMAL DISTORTION DATA HIDING

In this chapter, we present the minimum distortion data hiding technique. The chapter is organized as follows. The first section examines the feasibility of the data hiding for the compressed images. We introduce the definition of the hiding capacity as the difference between the transmission bitrate and the perceptual entropy of the signal. The definition is justified through an information theoretical derivation. Some experimental results regarding the viability of the data hiding option for the compressed images are given. These experiments examine the left-over redundancy in JPEG images.

The second section is on the design of the data hiding method. We first examine the requirements on the method. The method is introduced for the special case of three bit per block embedding. The details of the analysis and the optimization process are given for this special case. Since the method is scalable in compression and embedding bitrates, the analysis and the optimization results can be easily generalized. To develop more insight into the method, we present two analogies. These analogies are mostly interesting to the researchers aware of the related data hiding studies in the literature.

The third section discusses the general system operating at arbitrary compression and embedding bitrates. The details of the search for the optimum embedding parameters (the partitioning strategy) are given. An adhoc partitioning strategy is proposed and its performance is compared with the optimum strategy.

The fourth section presents an effort to further reduce the visibility of the noise. The goal of this section is to embed more hidden bits at the less perceptible regions of

the image and therefore distort the critical regions in lesser amounts. We use a human visual system model to estimate the contrast and the component masking effects of the DCT coefficients. The estimated values are used to weight the embedding error.

The fifth section presents some experimental results on the proposed method.

3.1 Hiding Capacity of the Compressed Images

We present a new interpretation for data hiding capacity which is in agreement with the existing works in the literature, [43, 13, 35]. You may consult [8] for details. The main claim can be stated as follows:

Claim: The amount of information that can be imperceptibly hidden in a media carrier is the difference between the bit rate used in the compression of it and the perceptual entropy of the signal.

A discussion of the perceptual methods in multimedia signal processing can be found in [29, 50]. Some applications of these ideas in the data hiding context have been given in [40, 52].

The theoretical approach to data hiding has been initiated by the recognition of the analogy between data hiding and the communication channel whose state information is only known by the transmitter [13]. This analogy has been extended by modeling the data hiding process as a game between the information hider and the attacker [35]. Some of the earlier work on the definition of the capacity also recognized the game-theoretic approach and resulted in simple, but elegant results [43].

Recently the duality between the source coding with side information at the receiver and data hiding (source coding with side information at the transmitter) has been explored [15, 3].

In here, we give a discussion of the claim above and establish the linkage between the claim and the other results given by Moulin *et. al.* and Chen *et. al.* [35, 13]. After visiting this relation, we evaluate some capacity bounds for different attacks.

3.1.1 System Model for Hiding

Before starting the discussion of the capacity problem, we give a model for the data hiding system. In Figure 7, the host signal and the hidden information are represented with $\tilde{x}$ and w respectively. These two signals may or may not be independent depending on the application. The first block of the transmitter is the perceptual source coder [29]. The output of this block represents the perceptually relevant components of the signals as represented by x_p and w_p. The next block in the transmitter is the data hiding block which combines x_p and w_p in a fashion that signal x is perceptually indifferent from $\tilde{x}$ and the hidden signal w is robustly protected from the attacks. Therefore, this block serves two purposes, imperceptible data hiding and attack compensation (channel coding). Attack on the composite signal which can be deterministic (compression) or random (additive noise) is represented by the channel. Finally, the channel output y is processed at the decoder to estimate x and w. Depending on the intended application, the host signal may or may not be available at the decoder. The latter case, which is known as blind data hiding, is more difficult for the decoder due to the influence of the host signal acting as an additional noise source.

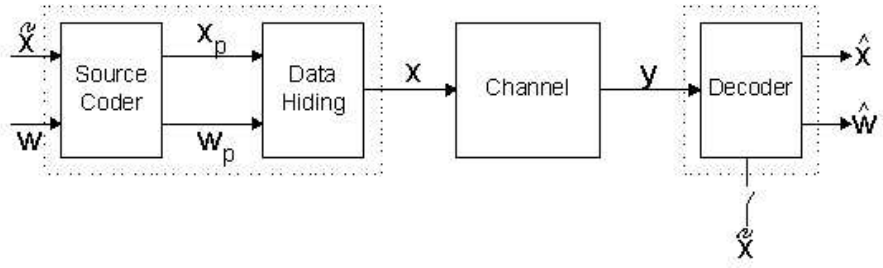


Figure 7: A System Model for Data Hiding

Data hiding problem with this model can be stated as the maximization of the rate of the signal w (R_w), under the maximum distortion constraint on x, while keeping the probability of extraction error of the hidden information ($P(\hat{w} \neq w)$) at an arbitrarily small value.

We present an example to clarify the details of the model. Let's assume that $\tilde{x}$ and w are text messages, that is $\tilde{x}[n]$ is the nth letter of a novel and w is the secret text message to be inserted in the novel. The distortion constraint on x allows us to change at most 1 letter out of 100 letters of the original text. A more frequent insertion (deliberate typo) will render the text to be useless. Under these circumstances, the first block of the encoder compresses messages to their essentials without any loss: that is, the redundancy of the language in its structure such as grammar, punctuation is removed. For example in English the letter q is always followed with the letter u. Therefore it is possible to remove all of the u letters coming after q's. Similarly all of the vowels in the novel can be replaced with dashes and an experienced reader should be able guess all the vowels. An important point is that after the perceptual compression, the information rate of text messages is reduced from $log_2(27)$ bits/letter (26 alphabet letters and space character) to R_x and R_w which is strictly less than $\log_2(27)$ bits/symbol. Data hiding block then constructs the composite message x from the perceptual messages x_p and w_p in a way that there is no ambiguity of message extraction at the decoder. The composite message passes through a proof-reader (attack channel) and reaches the hands of the intended party.

An important detail is that the rate of the signal x has to be at least $R_x + R_w$, because both signals x_p and w_p have to be combined together in an invertible fashion (the hidden information should be separable from the composite signal at the decoder). Therefore, if the alphabet of x_p has 2^{R_x} symbols and w_p has an alphabet size of 2^{R_w}, the composite alphabet has to have at least $2^{R_x+R_w}$ symbols, so that the composite signal can be partitioned into two components in a unique way.

Another point regarding the system is that the capacity of the attack channel should be at least $R = R_x + R_w$. Otherwise it is not possible to have a reliable communication between the input and output of the channel. We present an interpretation of the capacity conjecture based on this model in the next section.

3.1.2 The Claim

The equation for the capacity of an arbitrary channel is given as $C = \max_{p(w)}\{I(w;y)\}$ by Claude Shannon in 1948 [44]. In this equation w and y denote the channel input and output respectively. The channel is defined through an input-output map which can be probabilistic or deterministic (with probability distribution consisting of only 1's and 0's).

We first give the discussion for the *non-blind case*. The data hiding capacity C_h in this case can be written as follows:

$$\begin{aligned}
C_h &\overset{(a)}{=} \max_{p(w)}\{I(w;y|\tilde{x})\} \\
&\overset{(b)}{=} \max_{p(w)}\{H(w|\tilde{x}) - H(w|\tilde{x},y)\} \\
&\overset{(c)}{\leq} H(w_*|\tilde{x}) \\
&\overset{(d)}{=} H(w_*|\tilde{x}) + H(x_p) - H(x_p) \\
&\overset{(e)}{=} H(w_*|\tilde{x},x_p) + H(x_p) - H(x_p) \\
&\overset{(f)}{\leq} H(w_*|\,x_p) + H(x_p) - H(x_p) \\
&\overset{(g)}{=} H(w_*,x_p) - H(x_p) \\
&\overset{(h)}{\leq} (R_x + R_w) - H(x_p) \\
&\overset{(i)}{\leq} R - H(x_p) \\
&\overset{(j)}{\leq} C - H(x_p) \qquad (1)
\end{aligned}$$

Line (a) is the definition of the capacity for the non-blind case. Line (b) is the definition of the mutual information. The maximizing distribution is inserted in line (c) and the inequality is due to non-negativeness of entropy. In line (d), we introduce the variable x_p. Line (e) is valid since x_p is a function of $\tilde{x}$. In line (f) we use the rule that conditioning reduces entropy. Line (g) is the definition of the joint entropy. Line (h) follows from the Slepian-Wolf theorem (joint source coding [17, Theorem14.4.1]). Line (i) follows from the requirement of unique separation of the host data and the

hidden data. Line (j) is due to the assumption of reliable communication.

We see from the chain of inequalities that data hiding capacity for the non-blind case is upper bounded by the difference of the capacity of the attack channel and perceptual entropy of the host signal, as conjectured.

The *blind case* is more difficult to analyze, but recent studies have established important steps in this direction. In [35], data hiding operation has been defined as a game between the hider and the attacker. If the optimum strategy for both players is exercised, the capacity of the data hiding game is given by $C = \max_{p(x,u|\tilde{x})} \min_{p(y|x)} (I(U;Y) - I(U;\tilde{X}))$. The composite signal x is constrained to be below a distortion limit. The attacker also has a maximum distortion limit which prohibits the use of excessive distortion on x. In this theory the variable u is represented as the auxiliary variable, or as a dummy variable, over which maximization is accomplished. We believe that the signal u has an important role in the data hiding context. We propose to interpret the signal u as the signal x_p which represents the perceptually coded version of the signal $\tilde{x}$ according to our model.

Assuming that we have fixed the attack channel ($p(y|x)$), the capacity in this case can be written as follows:

$$\begin{aligned} C_h &\overset{(a)}{=} \max_{p(x,\,u|\tilde{x})} (I(U;Y) - I(U;\tilde{X})) \\ &\overset{(b)}{\leq} \max_{p(x,u|\tilde{x})} I(U;Y) - \min_{p(u|\tilde{x})} I(U;\tilde{X}) \\ &\overset{(c)}{=} \max_{p(x,x_p|\tilde{x})} I(X_p;Y) - \min_{p(x_p|\tilde{x})} I(X_p;\tilde{X}) \\ &\overset{(d)}{=} C - H(x_p) \end{aligned} \tag{2}$$

Line (a) is the definition of the capacity for a fixed attack channel. In line (b), we upper bound the equality in (a) by maximizing the two terms of (a) independently. In line (c), we make the analogy of identifying u with x_p. The second term of the line (d) can be recognized as the minimum rate that is necessary to construct signal x_p from $\tilde{x}$, which is the entropy of the signal x_p (perceptual source coding). The first

term of line (d) is the definition of the capacity of the attack channel (channel coding) for the signal x_p.

The proposed analogy can be viewed as follows: the host signal is first coded to the signal u ($\tilde{x} \rightarrow u$) and then the signal u is coded once more to the signal x ($u \rightarrow x$). The final signal is transmitted through the attack channel. For data hiding applications, attacker has to watch the perceptual quality of the resultant signal after the attack. Because of this, attack tools can be pictured as tools operating on the perceptual components of the host signal or they can be visualized as the operators working in the perceptual domain. With this visualization, the first coding operation, from $\tilde{x}$ to u, can be thought as the perceptual source coding (projection operation of the signal $\tilde{x}$ to the domain of the attack tools). The second coding operation, from u to x, is the channel coding for a particular attack tool (transformation of the signal x_p to the signal x whose components lie in the range space of that particular attack). With this analogy, the maximum value of the expression $I(U;Y) = I(X_p;Y)$ represents the maximum rate of reliable communication of the perceptual components of the host signal. We emphasize that if the attacker could apply arbitrary attacks, the analogy proposed would not be valid, since there would not be a common domain for the attacks.

3.1.3 Capacity Achieving Conditions

We list the capacity achieving conditions for the two cases of data hiding. We start with the non-blind case. The requirements can be listed as follows: 1. There is a small probability of error at the decoder (ignored term in line (c) is bounded by the Fano's inequality [17, Lemma 8.9.1]) 2. The hidden information should depend only on x_p i.e. signals $\{x, x_p, w_*\}$ should form a Markov chain of $x \rightarrow x_p \rightarrow w_*$ (from line f). 3. The perceptual source coder should be perfect (line h). 4. The data hiding operation should be invertible (line i). 5. Hidden data should be embedded at the maximum rate allowed by the attack channel which is $R_w = C - R_x$ (line j).

All the requirements, other than the second one, emphasize the ideal operating conditions for the data hiding system. The second requirement says that to maximize the capacity, the hidden data should be in relation with the perceptual components of the host signal, but not with the host signal itself.

For the blind case, line (b) implies that the data hiding throughput C_h is maximized when the $p(u|x)$ appearing in both terms of line (b) are the same (The probability distribution $p(x_*, u_*|\tilde{x}) = p(x_*|u_*\,\tilde{x})p(u_*|\tilde{x})$ maximizes the first term and at the same time the distribution $p(u_*|\tilde{x})$ minimizes the second term). If the analogy between u and x_p is applicable, we expect this relation to be satisfied (perceptual coding is independent of the channel).

3.1.4 Capacity Estimates Under Different Attacks

In this section we present the capacity estimates of data hiding systems under some practical attacks. We experiment with the 512x512 Lena image whose pixels are represented with 256 gray levels. To determine the perceptual entropy of the Lena image, we used Watson's human visual system model and assumed that pixels below the just noticeable distortion (JND) threshold do not contribute anything perceptually [50].

Noiseless Channel: If the attack channel is noiseless (no-attack condition), the capacity is given by $C(D) = 8 - R(D)$ bits / pixel, where $R(D)$ is the perceptual rate distortion function [50]. The top panel of the Figure 8 shows the percentage of the pixels available for data hiding ($C(D)/(512 \times 512)$) as the allowable distortion due to the embedding increases. These pixels (or the transform coefficient in DCT domain) have the values below JND. They can be discarded according to the Watson's model. The Lena images at different allowable distortion levels (no distortion, JND, 3 $\times$ JND, 20 $\times$ JND) are shown in Figure 9. From these pictures and from the graph portraying the change of available coefficients for hiding, we can say that as the JND tolerance is multiplied, we achieve more room for hiding.

JPEG Compression Channel: The JPEG compression operation is inserted in the attack channel. The bottom panel of Figure 8 shows a trend similar to the one in the top panel. It is clear that JPEG compression takes most of the redundancy, but the left over redundancy is enough to insert hidden data, without any perceptual distortion, at 1310 pixels of the Lena image corresponding to the 0.5% of total number of pixels. It is important to note that Watson's model provides us a method to calculate the distortion in another unit (JND units). We examine the distortion at the different multiples of JND and check the visibility of the distortion at those levels. The number of available hiding coefficients at those distortion values are the coefficients whose values are below the JND multiple minus the number of coefficients discarded by the JPEG algorithm. Our goal is to examine the efficiency of the JPEG compression. We would like to examine whether JPEG can discard more data or not without any perceived quality loss. The coefficients that are kept by JPEG which do not pose any visible distortion when discarded can be utilized for hiding. The x-axis of the Figure 8 is the tested quanta of JND expressed in PSNR. We have preferred to use PSNR, not the JND value in this axis; since the PSNR values of Lena image can be better interpreted than the JND values (because of the decades of the compression research on this image). The tested distortion values in Figure 8 range from 1xJND to 3xJND (inclusive) with 0.1 JND increments.

In Figure 9, we show the original Lena image and its distorted versions to observe hiding distortion. The distorted images are the quantized versions of the Lena image at the multiples of JND. As expected, the hiding capacity increases if we allow more distortion. The point where the introduced distortion turns to intolerable, the hiding capacity of the image. We would like to point out that this cross-over point can not be determined objectively.

Additive Noise Channel: An attack of a binary symmetric memoryless channel with the transition probability of ϵ functioning independently on each transmitted bit

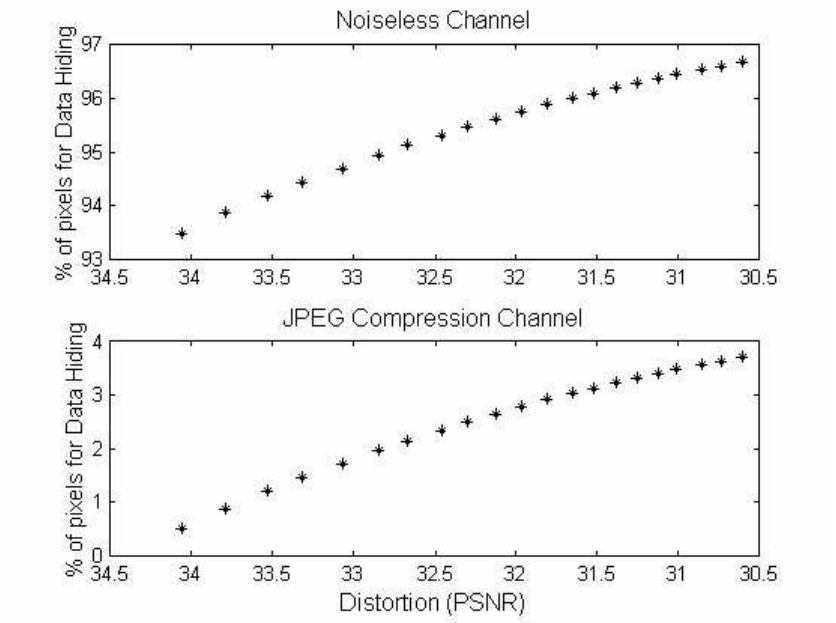


Figure 8: The top panel shows the data hiding payload of Lena image as the distortion due to the embedding increases gradually. The lowest distortion value of the Figure corresponds to the 1xJND distortion which is 34.05 dB in terms of the PSNR metric. The hiding capacity is given by the percentage of the total number of transform coefficients which can be modified for hiding. The bottom panel gives the capacity estimate of the same image under the JPEG compression attack.

of Lena image is assumed. The capacity in this case is given as $C(D) = 8(1-H(\epsilon)) - R(D)$ bits/pixel when this value is greater than zero; otherwise zero. According to the vision model adopted, the Lena image can be compressed at 0.52 bits/pixel without a perceptible distortion. Therefore for the $\epsilon = 0.25$ it is possible to encode almost 1 bit of hidden data per pixel without a perceptual quality loss.

Image rotation, flipping and other invertible operations: An invertible attack on a signal does not change the entropy of the signal. Therefore any invertible attack such as image flipping or rotation does not pose a threat to the capacity. But in practice, undoing the effects of these operations (especially sequential combination of these operations) can be computationally very difficult and perhaps impossible in practice. The field of cryptography is built upon on this premise.

Figure 9: Lena images with different levels distortion are shown. The top left image is the original. The top right image is distorted up to the 1xJND threshold with the hiding payload of 1310 pixels. The bottom left image is at 3xJND distortion with the hiding payload of 9670 pixels. The bottom right image is at 20xJND distortion with the hiding payload of 15000 pixels.

3.2 Minimum Distortion Data Hiding Technique

The minimum distortion data hiding algorithm is introduced in this section. The design is first given for the special case of three bit per block embedding is given and then generalized.

Our goal is to build covert communication schemes for the encoder-decoder devices of the existing communication systems. The JPEG standard is also one of the common communication standards. It is a building blocks of the MPEG standard. The outline of the algorithm presentation is as follows:

Analysis: We determine the quantization error after the hiding process given the quantization error before hiding (only compression). For a fixed partitioning strategy, distortion per bit embedding is determined. Optimal partitioning strategy for 2 bits per block embedding at JPEG Quality Factor=80 is determined.

Search For Best Partitioning Strategies: Best partitioning strategies at different embedding and compression levels are found by an exhaustive search. An ad-hoc partitioning strategy closely tracking the performance of the best partitioning strategy is proposed.

HVS Guided Embedding: A human visual system guided embedding system is proposed.

Subjective Tests: The results of the subjective tests are discussed. Some other experiments on image resolution, file length are also described.

3.2.1 Requirements

The requirements from the data hiding algorithm to implement an application on top of the existing image communication standard are listed as follows:

System Compatibility: The method should be compatible with the existing image compression standards (JPEG or MPEG). Our goal is to implement hiding based applications that are not directly supported by the standards.

System Blindness: The hidden data decoder located at the receiving end of the channel (customer premises) should operate in the absence of any additional information such as the transmitted image, embedding parameters etc.

System Efficiency: The decoder of the hiding algorithm should be simple enough that the hidden data decoder can be easily and cost-effectively integrated to the existing JPEG image decoder. The encoder, which is located at the service provider end of the channel, can have more computational complexity.

Scalability: Image sequences are compressed at a variety of different compression levels depending on their content. Similarly, the hiding bitrates may vary significantly from application to application. The proposed hiding solution should be designed to be scalable in compression-embedding bitrates.

Minimum Distortion: The solution should be designed to minimize the perceived distortion in the after hiding image.

3.2.2 Abstract Description of the Method

We present a simple presentation to show how the designed method works. This presentation illustrates the codeword partitioning idea for a 2 dimensional transform domain coding system.

JPEG Compression: JPEG compression has four stages:

1. Dividing the image into 8×8 blocks,
2. Calculating the DCT of every block,
3. Quantization of the DCT coefficients,
4. Run-length and entropy coding of the quantized coefficients.

The third step is the most critical step. At the third step, the coefficients with higher significance are quantized finely and the coefficients with lower significance, or with smaller contribution to the perceived quality, are quantized coarsely. The final step of entropy coding effects the file length but not the quality of the image.

The filter banks with their special data structures (significance trees) proved to be more efficient at image coding than the JPEG algorithm [45, 48]. JPEG enjoys a lower computation complexity (due to the fast DCT implementations), a lower memory requirement (due to the sequential step by step operation) and a parallel processing implementation possibility (processing of 8x8 blocks). On the other hand its compression performance in terms of bits per pixel at a given PSNR is inferior to the one of filter bank approaches. The deployment level of JPEG and the abundance of transmission bandwidth in today's market conditions makes the switch from JPEG to the filter bank based approaches unlikely. This belief is further strengthened by the adoption of the JPEG based methods in the video standards.

We present an abstract description for a JPEG-like quantization system. This presentation illustrates how JPEG operates at 2 dimensions. The hiding functionality is included to the system at the very step. Readers may consult [30, chapter 12] for the details of transform coding.

In Figure 10, we give an illustration of the JPEG operation, [30, ch. 12.2]. The Figure 10a illustrates the structure of the signal to be compressed. For the sake of simplicity, we have assumed the signal to be compressed is a vector of length two whose components are correlated in such a way that their values fall in the ellipse shown. The transform stage at Figure 10b rotates the coordinate axes to decorrelate the signal. The decorrelated coefficients are quantized depending on their significance levels. Quantization provides a systematic way of mapping the input to the nearest codeword. As can be seen from Figure 10c, the major axis of the ellipse is longer than the minor axis. Implying that the information content of the major axis is higher. Due to the significance of major axis more codewords (fine quantization) are assigned to this axis (The major and minor axes are proportional to the variance of the rotated random variables).

The compression problem can be posed as a basis limitation problem [28]. A 8x8 matrix have 64 different independent basis functions. The basis limitation problem is finding a lower dimensional subspace approximating the given signal with the minimum mean square error. In the example given, the input vector has two dimensions. The best one dimensional limitation for this vector is the direction of major axis (This is equivalent to the KL transformation when interpreted statistically, [28, page 166]). In other words, the axis with the lower information content is discarded through the basis limitation operation.

In Figure 10d, you can see the complete set of compression codewords covering the ellipse. Compression operation is the mapping of the input signal to the nearest codeword which is the tip of the vectors shown.

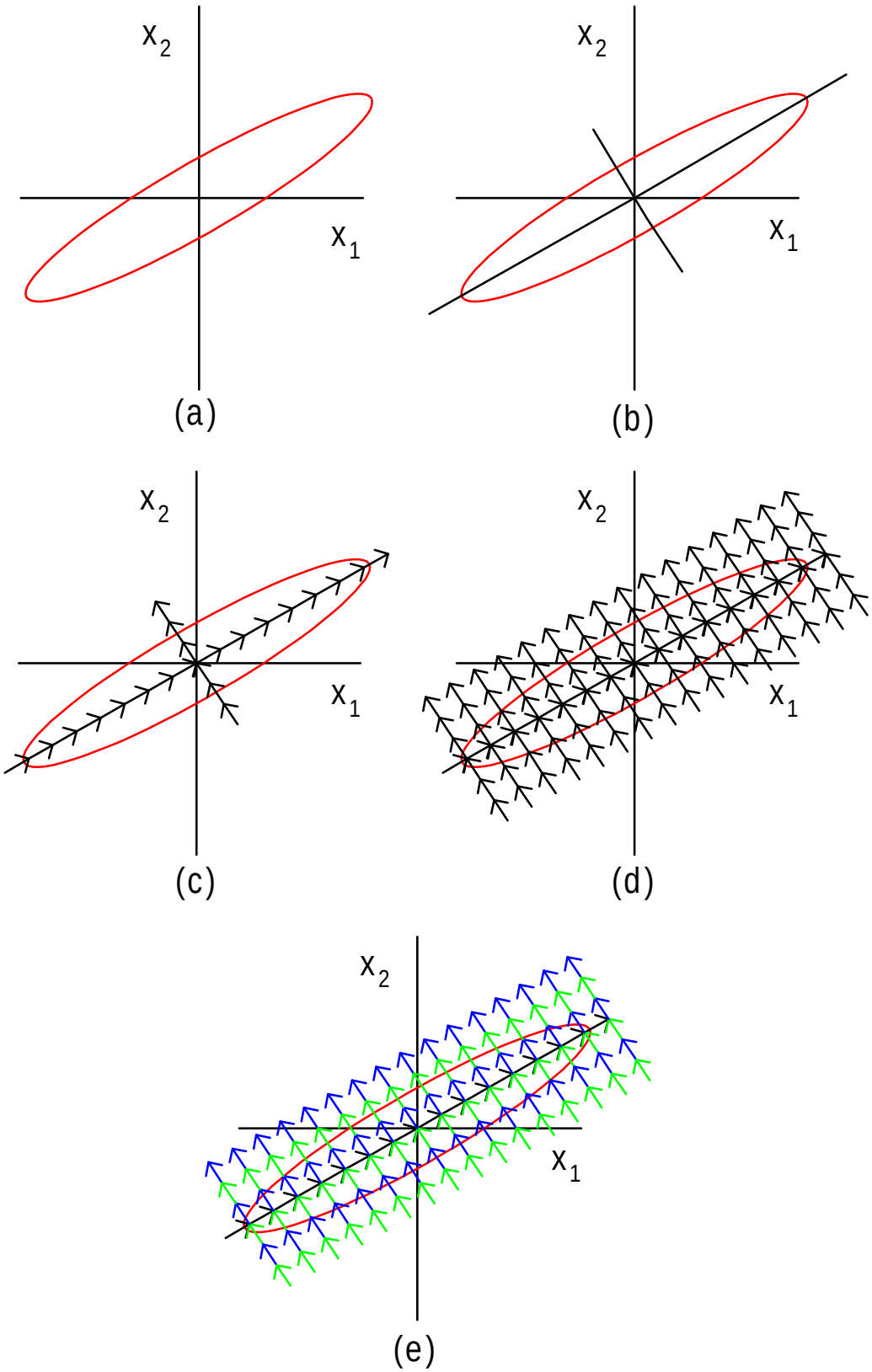
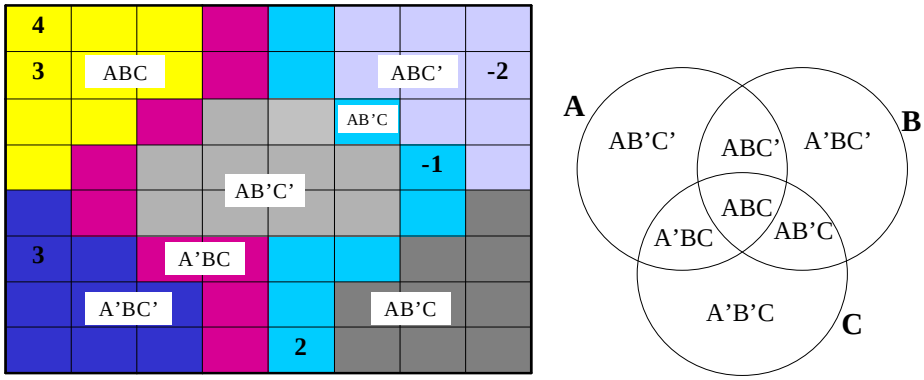


Figure 10: JPEG Codewords



(a) An 8×8 Data Block in DCT domain (b) Partitions

Figure 11: Partitioning of an 8 by 8 block

In Figure 10e, we present the same codeword space with two different colors. The purpose of using two colors is to partition the codeword set into two disjoint sets. If the partitions are known by the encoder and decoder, the encoder can send one bit of hidden information by the color code. If the hidden information is blue, the transmitted word is the nearest blue-colored codeword to the original word.

JPEG transform works in the described manner. JPEG uses 8x8 blocks, therefore it has 64 dimensional vectors as input. Our research project involves the determination of good partitioning strategies for JPEG codewords.

3.2.3 Hiding Method

The method is examined on a special case of hiding three bits per block. The results are generalized in the next section. We assume that the encoder and decoder pairs agree on the hiding method beforehand. The decoder uses a-priori knowledge on the method and the received JPEG image to extract the hidden information. We start with the decoder first.

3.2.3.1 Hidden Data Decoder

The decoder receives a JPEG image known to have 3 bits of hidden data per block. The image decoder applies the conventional JPEG decoding techniques to decode the

image. We do not explain the operation of conventional JPEG decoding in here. The hidden data decoder generates the embedded bits as follows:

The hidden data decoder operates on the quantized DCT coefficients. The values in the 8×8 block in Figure 11a are the quantized DCT coefficients. The value of 4 at the upper left hand corner of the block shows that the DC value of the block is dequantized to the 4 times the DC quantization step size.

The quantized DCT coefficients are interpretted as the JPEG codewords. The designed hiding method should partition the 64 dimension codeword space into $2^3 = 8$ disjoint sets to embed 3 bits per block.

To partition the JPEG space, the block in Figure 11a is divided into sets, each shown with different colors. These sets are also shown in Figure 11b for illustration purposes. The hidden bits are calculated from the quantized DCT coefficients via a linear combination operation. For the discussed three bit hiding example, the following summations are evaluated:

$$\begin{aligned} \text{Bit-A} &= \sum_{x \in A} C_{ij} = 6 \equiv 0 \,(\text{mod } 2) \\ \text{Bit-B} &= \sum_{x \in B} C_{ij} = 8 \equiv 0 \,(\text{mod } 2) \\ \text{Bit-C} &= \sum_{x \in C} C_{ij} = 8 \equiv 0 \,(\text{mod } 2) \end{aligned} \tag{3}$$

The system is operating blindly if the coefficients appearing in the above summations are known beforehand. Finding the assignment strategy of the coefficients to the equations constitutes the partitioning problem. We can rewrite the equation (3) in the matrix form as follows:

$$\begin{bmatrix} \text{Bit-A} \\ \text{Bit-B} \\ \text{Bit-C} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ 0 & 0 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} (c_1)_2 \\ \vdots \\ (c_{64})_2 \end{bmatrix} \tag{4}$$

The columns of the matrix include at least a single occurrence of "1". The double or triple occurrences of "1" indicate that the element appears in more than one hidden bit extraction equation, i.e. it is in one of the intersecting sets of Figure 11b. The partitioning operation is the determination of the assignment matrix. For the described special case, an arbitrary partitioning strategy is used for illustration purposes.

Embedding Options: The next point is the examination of the structure of the partitions. We would like to examine the neighborhood structure of the codewords.

In the described example, the extracted bits are $(0, 0, 0)$, see (3). The hidden bits can be changed to $(1, 0, 0)$, if an element of the set $\{AB'C'\}$ is incremented or decremented. This can be done in $9x2 = 18$ different ways.

It can be seen that every other combination of hidden bits around the word $(0, 0, 0)$ is accessible by a single increment/decrement operation. This shows that the minimum distance between partitions is 1.

The distance 2 and the distance 3 neighborhood around the codewords can also be calculated similarly. This gives us following neighborhood information:

1. There are $18 = 9 \times 2$ codewords of different partitions at distance 1 around every codeword.

2. There are $486 = 9^2 \times 3 \times 2$ codewords of different partitions at distance 2 around every codeword.

3. There are $1458 = 9^3 \times 2$ codewords of different partitions at distance 3 around every codeword

The distance between words coincides with the number of transform coefficients needed to be modified to embed the intended hidden bits. We have seen that there are around 2000 different options to embed a hidden bit combination given any input word. The encoder should search the option set to find the option with the least distortion cost.

The decoder of the system indirectly constructs the encoder. As can be seen from the abstract description, the decoding function checks the color of the partition and the encoder moves the given word into the right colored set. The encoder searches for a favorable word with the desired color to move the input word to.

3.2.3.2 Error Correction Interpretation

We present the following error-correction analogy. This section is mainly interested for the readers looking for additional insight into the design. It can be skipped without any discontinuity.

The decoding operation in (4) are the binary sums of the selected components on a 64 dimensional vector. This operation and the matrix can be interpreted in the error-correction context.

In the error-correction coding terminology, the matrix in (4) is called parity check matrix. The equations are called parity check equations. The result of the parity check equations is called syndrome. The set of codewords with the same syndrome are called a coset.

In the data hiding context, we have called the matrix in (4) as the decoding matrix. The equations are called as hidden bit extraction equations. The result of the hidden bit extraction equations is called the hidden bits. The set of codewords with the same hidden bit combination is called a partition.

We present a simple (2,5) error-correction code to show the analogy with an example. The 32 possible words are partitioned into 8 sets by the cosets. The codewords and the syndrome information is given in Table 3.

The data hiding encoder works as follows: Given a hidden bit combination and an input word, the encoder searches the row of the Table 3 whose coset label is the same as the given hidden bit combination. The word with the minimum distance to the given word is the output of the hidden data encoder. The decoder evaluates

the parity bit generation equations to generate the syndrome. The syndrome is the embedded bits. The core of the research is on the partitioning of the codeword space.

Table 3: Error-Correction Interpretation: The standard table of a simple 1 error correcting code. Data Hiding Interpretation: Look-up table for decoding three hidden bits (syndrome).

Syndrome	Received words			
000	00000	00111	11110	11001
100	10000	10111	01110	01001
010	01000	01111	10110	10001
001	00100	00011	11010	11101
111	00010	00101	11100	11011
110	00001	00110	11111	11000
101	01010	01101	10100	10011
011	01100	01011	10010	10101

3.2.3.3 Information Theoretic Interpretation

An information theoretical interpretation of the described decoder is given in this section. Readers should consult [35, 13, 2] for more information on data hiding theory.

A standard tool of information theory is the concept of *typicality*. The proofs of the channel coding theory (error coding) and the rate-distortion theory (source coding) heavily depends on the typicality of the sequences [1].

We quote the following lines from [14]. At this part of the paper, Chen *et. al.* discuss the optimality of the data encoding:

1) Hidden QIM: As we show in this subsection, one can achieve the capacity by a type of "hidden" QIM, i.e., QIM that occurs in a domain represented by the auxiliary random variable u. (...) u is randomly drawn from the i.i.d. distribution p_u, which

[1] An informal definition of typicality can be given as: A realization of a random process is typical if the "statistics" derived from that realization "matches" the "statistics" of the ensemble, see [38, 17] for more info.

is the marginal distribution corresponding to the host-signal distortion p_x and the (capacity) maximizing conditional distribution $p_{u,e|x}$. (...)

QIM embedding in this u-domain corresponds to finding a vector u_0 in the mth quantizer's codebook that is jointly distortion-typical with x and generating $\mathbf{e}(u_0, x)$.

Chen uses the typicality of the sequences to define a encoder/decoder pair. The operation is built on random variable u, which we would like to think as the u domain.

The encoder (in the noiseless case) finds the codeword of the mth codebook which is jointly typical with the input. The operation takes place in u domain. The decoder checks all words to find the word which is jointly typical with the received word. The codebook index (m) is the hidden information.

In our method, the encoder searches a partition of the codebook to find the codeword in minimum distance to the given word. The operation takes place in DCT domain. The encoder has to check many possibility to determine the best embedding option. The decoder evaluates hidden bit extraction equations to extract the hidden bits. In other words, the concept of typicality check is substituted with the described best option search operation in a partition.

3.2.3.4 Hidden Data Encoder

The encoder is indirectly determined by the decoder. The encoder modifies the JPEG image to cast the hidden bits. The encoder is located at the transmitter site. Therefore it has the original image information (uncompressed image) and additionally it has the information on the local image properties. Encoder with these additional information sources (the original image and the human visual system estimates) can search through the different embedding options to find the best way of modifying the transform coefficients. In other words, the encoder searches the codeword space around a given JPEG codeword to determine the least distortive element of a partition determined by the hidden bits.

We illustrate the encoding process explicitly in Figure 12. In Figure 12, the original DCT coefficients and JPEG quantization step sizes are shown. JPEG compression rounds the original DCT coefficients to the nearest multiple of a step-size and the multiplicity of the step-size is the quantized coefficient. Without the hiding function, the quantized coefficients transmitted to the other end of the channel (after entropy coding).

The hiding function modifies the quantized coefficients to accommodate the hidden bits. In the example shown, the block is partitioned into 3 sets shown with distinct colors. At the shown scenario, the encoder is given two options to embed the hidden bits. The option 1 is a single increment/decrement of a coefficient in the first row of the block (the row with the red color). The option 2 is a similar modification for a coefficient in the rows 2 and 3. The encoder calculates the distortion due to different options and selects the one with the minimum value. As can be seen from this example, the quantization step-sizes are the main factors effecting the distortion.

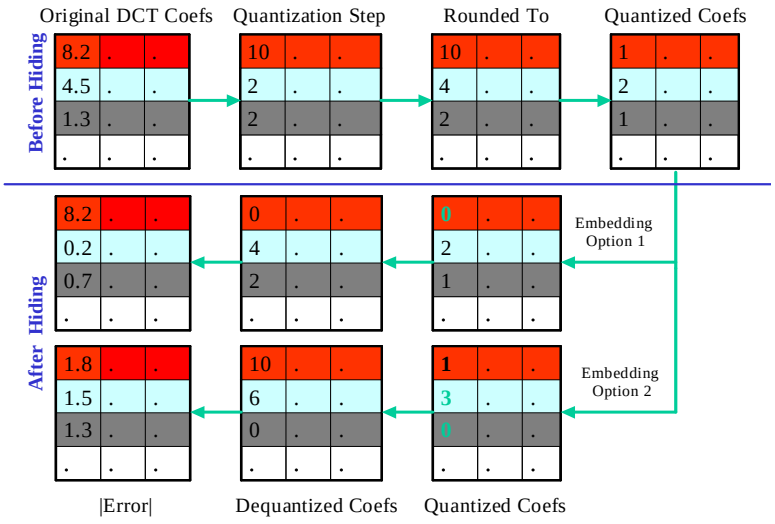


Figure 12: An Illustration of the Encoding Options

In this study, we use MSE and $\mathcal{L}_1$ metrics with their HVS weighted counterparts to measure the distortion. The cost of each embedding option according to the selected measure is calculated and the one with the minimum value is selected. It should be

noted that the decoder does not need to know anything about the selected distortion measure or the original DCT coefficients or the particular option chosen to extract the hidden bits.

It is clear from Figure 12 that the transform coefficients with low quantization step sizes introduce less distortion at their value increment/decrement. The assignment process of the right transform coefficient to the right subset is critical to reduce the overall distortion. We give an analysis of the partitioning operation in the following chapters.

3.2.3.5 A Fast Search Method for the Encoder

In this section, we present a fast method for the embedding option search process.

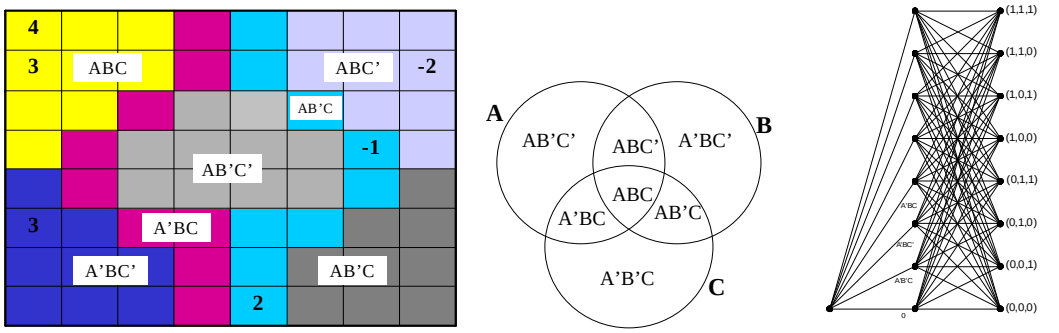
The number of embedding options can be overwhelming. For the example given earlier, the number of cases that has to be compared at every block of embedding is more than 1500. In this section we present a method to reduce the geometrically increasing number of cases to a linearly increasing one. This reduction only applies to the difference based distortion metrics such as MSE, $\mathcal{L}_1$ and their weighted versions.

We introduce the set-leader term in this section[2]. As before, the block given in Figure 13 is partitioned into 7 sets each with 9 elements. A specific element of every set is labeled as the set-leader.

The set-leader is the element whose value increment or decrement has the least algebraic difference from the original (uncompressed) value.

We give a simple example to illustrate this election. The set-leader of $\{ABC\}$ (yellow colored partition) can be determined as follows: The yellow colored set has two non-zero quantized coefficients which are 4 and 5. Assume that the quantization step sizes of these coefficients are 5 and 6 respectively. After de-quantization, these coefficients are mapped to 20 and 30 respectively. If we assume the original transform

[2]Coset leader is the analogous terminology appearing at error-correction coding.



(a) An 8×8 Data Block in DCT domain

(b) Subsets

(c) Search Trellis

Figure 13: Partitioning of an 8 by 8 block, Three Sets and Search Trellis

coefficients as 22 and 32, the algebraic error is 2 for both coefficients. If we change the value of the first coefficient from 4 to 5, the de-quantized signal becomes 25 and the new error on this coefficient becomes $25 - 22 = 3$. Similarly a increment on the other coefficients causes an error of 6 units. These calculations of error evaluated for every element of the set once and the element with the least algebraic error of is declared as the set leader.

After the election of set-leaders, the cost of different options to embed hidden bits is evaluated through only the set-leaders. For difference based metrics, it is clear that the non-leader elements cause more distortion than the set-leaders. Therefore search process does not need to cover these cases. This observation provides us the reduction in search complexity.

Continuing with the given example. If the correction that we need to make is $(1, 1, 1)$ (the encoder has to change one coefficient in the sets $\{A\}$, $\{B\}$ and $\{C\}$) then this can be accomplished through the following four possibilities:

1. Increment/Decrement the value of the set-leader of $\{A, B, C\}$, i.e. $(1, 1, 1)$.

2. Increment/Decrement the value of the set-leaders of $\{A, B, C'\}$ and $\{A', B', C\}$, i.e. $(1, 1, 0) + (0, 0, 1)$

3. Increment/Decrement the value of the set-leaders of $\{A, B', C\}$ and $\{A', B, C'\}$,

i.e. $(1,0,1)+(0,1,0)$

4. Increment/Decrement the value of the set-leaders of $\{A',B,C\}$ and $\{A,B',C'\}$, i.e. $(0,1,1)+(1,0,0)$

5. Increment/Decrement the value of the set-leaders of $\{A,B',C'\}$ and $\{A',B,C'\}$ and $\{A',B',C\}$, i.e. $(1,0,0)+(0,1,0)+(0,0,1)$.

Therefore after the election of 7 set-leaders, the search process reduces to comparison of 4 options. More generally, the simplified search can also be done through a trellis as in Figure 13. The target state of the trellis is the correction move needed to embed the intended hidden bits. The initial state is the state of no correction $((0,0,0))$. The metric on the branches is the cost of the modification of the set-leader establishing the desired the correction between two nodes. The horizontal branches have zero cost.

3.2.4 Analysis of Partitioning

In this section, we present an analysis of the partitioning operation. We examine the simplest non-trival case of embedding 2 bits per block. The other cases can be generalized from this case.

For the 2 bit embedding case, the 64 transform coefficients of an 8×8 block is divided into 3 sets. The binary sum of the elements in the sets $\{A\}$ and $\{B\}$ are the hidden bits. These sets are shown in Figure 14.

The partitioning operation is the assigning the DCT coefficients to the subsets. For example, a strategy for the set $\{AB\}$ can be the placement of the DC coefficient and 5th, 7th and 9th AC coefficients in this set. In this section, we quantify the goodness of different strategies.

The process resembles resource allocation or logistics problems at which resources are shared between sub-functions for the optimization of the main function.

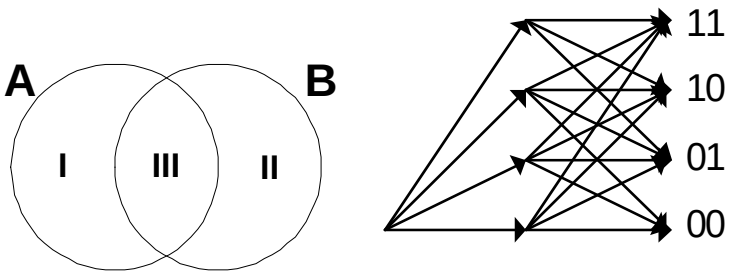


Figure 14: The Sets and The Cost Minimization Trellis for 2 bits per block Hiding

Quantization step sizes used at JPEG compression is available to both encoder and decoder. We would like to make use of this information at the determination of best partitioning strategies. It should be noted that this depency does not damage the blindness of the system if a deterministic partitioning algorithm can be given on the quantization steps. If this is the case, the encoder-decoder can run the partitioning algorithm at the two end of the channel given the quantization table.

The default quantization step sizes of the JPEG system are not the same for all 64 transform coefficients. The coefficients with more variance (major axis of the ellipse in Figure 10) are finely quantized, the coefficients with smaller variance are transmitted only if its value is far from its expected value (minor axis of the ellipse in Figure 10). Below is the default JPEG quantization matrix at quality factor of 80%:

$$\text{Default JPEG Quantization Matrix at Quality Factor 80\%} = \begin{bmatrix} 6 & 4 & 4 & 6 & 10 & 16 & 20 & 24 \\ 5 & 5 & 6 & 8 & 10 & 23 & 24 & 22 \\ 6 & 5 & 6 & 10 & 16 & 23 & 28 & 22 \\ 6 & 7 & 9 & 12 & 20 & 35 & 32 & 25 \\ 7 & 9 & 15 & 22 & 27 & 44 & 41 & 31 \\ 10 & 14 & 22 & 26 & 32 & 42 & 45 & 37 \\ 20 & 26 & 31 & 35 & 41 & 48 & 48 & 40 \\ 29 & 37 & 38 & 39 & 45 & 40 & 41 & 40 \end{bmatrix} \quad (5)$$

Our task in this section is determining a good strategy to map 64 coefficients in equation (5) to the sets in Figure 14. We introduce the following notation:

$$\begin{aligned} Q &= \{q_1, q_2, \ldots, q_{64}\} \\ S_I &= \{q_1^I, q_2^I, \ldots, q_k^I\} \\ S_{II} &= \{q_1^{II}, q_2^{II}, \ldots, q_l^{II}\} \\ S_{III} &= \{q_1^{III}, q_2^{III}, \ldots, q_m^{III}\} \end{aligned} \tag{6}$$

In equation 6, Q denotes the increasing sequence of quantization step sizes (the elements of matrix in (5)). The sets S_I, S_{II} and S_{III} are the partitions of Q which are ordered in the same manner.

An equilibrium argument is used to check the optimality of a given solution. A strategy is called optimum if an interchange of two elements between two partitions increases the distortion. This condition is called the saddle point condition in the game theory literature, [36].

We describe the overall distortion as follows:

$$D = D_I U_I + D_{II} U_{II} + D_{III} U_{III} \tag{7}$$

In the equation above, D represents the overall distortion of the embedding system. D_I represents the average distortion upon the usage of S_I to embed data and U_I represents the percentage of usage of set S_I for embedding, that is the probability of changing an element of set S_I for data embedding purposes. We call this probabilistic value as the utilization factor of set S_I, [36].

First, we analyze the derivation of D_I for a given S_I. Next, we present a discussion on U_I and then conclude this section by finding the optimum strategy for 2 bit embedding.

3.2.4.1 Effective Elements

In this section, we show that some elements of a set can never be selected as the least distortion causing coefficient. In other words, some members of a set can never be selected as the set-leaders, please see section 3.2.3.5 for the definition of the set-leader.

First, we explain the intuition behind this result. Assume that two transform coefficients with step sizes of 5 and 50 are assigned to the set S_I. If the element with the quantization step size of 5 is chosen as the set-leader, the *worst* case distortion by moving its value by one step-size higher or lower is 5 (The worst case occurs when the coefficient has zero error before hiding). On the other hand, if the element with the quantization step size 50 is selected as the set-leader, the *best* case distortion by moving its value by one step-size higher or lower is 25. Therefore, if these elements reside in the same set, the one with the higher quantization step size can never be elected as the set-leader.

In Figure 15, the distortion before and after the hiding process is shown. In this picture, the codewords are shown with the circles and the original transform coefficient is shown with a cross. The compression operation maps the original transform coefficient to the nearest codeword, which is the codeword zero in Figure 15. If this coefficient is to be modified by data embedding function, the least distortive action is moving the codeword one step to the right. Distortion values before and after hiding are shown by D_b and D_a in Figure 15.

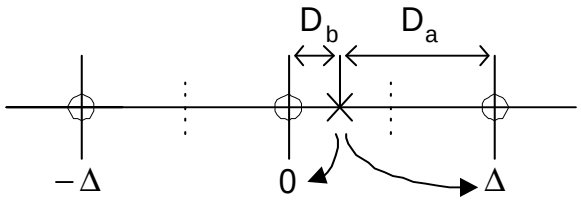


Figure 15: Codewords Before and After Data Hiding

The quantization error (distortion before hiding) is solely due to rounding operation. It can be seen by inspection of Figure 15 that the distortion after hiding is $[\frac{\Delta}{2}, \Delta]$ as shown in Figure 16.

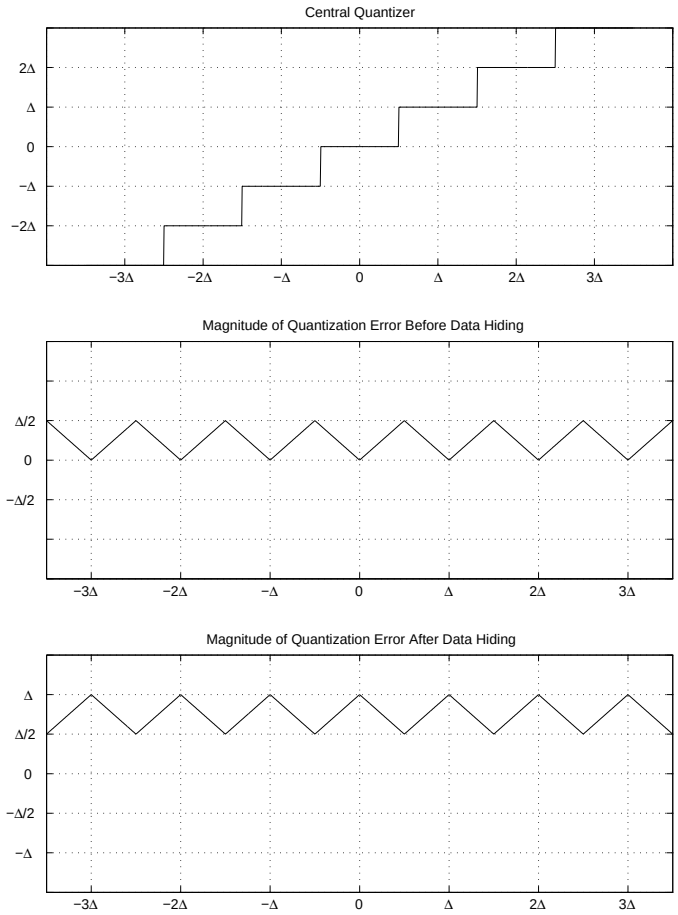


Figure 16: Illustration of the Distortion Before and After Data Hiding

For the elements of set S_I, the best and worst case distortion after the hiding operation is:

$$\begin{aligned} \min \text{error}\{S_I\} &= \left\{\frac{q_1^I}{2}, \frac{q_2^I}{2}, \ldots, \frac{q_k^I}{2}\right\} \\ \max \text{error}\{S_I\} &= \left\{q_1^I, q_2^I, \ldots, q_k^I\right\} \end{aligned} \tag{8}$$

The values of the maximum error set corresponds to the case of zero-error before hiding (The original data coincides with a codeword). The data hiding function seeks

the minimum of the given possibilities. Therefore, the worst-possible error is

$$\min \max \text{error}\{S_I\} = q_1^I. \tag{9}$$

Similarly, the best possible cases for the hiding distortion are listed in the $\{minerror\}$ set (The original data is one half step-size away from a codeword).

If the element j is selected as the set-leader, its error should be less than or equal to the worst-case error of the system (q_1^I). Or in other words, if the error on a coefficient is larger than q_1^I, it can not be the set leader. The minimum error for the element j is q_j^I is $\frac{q_j^I}{2}$. Therefore if $\frac{q_j^I}{2} > q_1^I$, q_j^I can not be the set-leader.

We introduce a new terminology for the potential set-leaders: The effective elements of a set are the elements that can be selected as the set-leaders. The effective elements (q_k^I) of a set (S_I) satisfy the following condition:

$$\text{Condition for Effective Elements:} \qquad q_k^I \leq 2q_1^I \tag{10}$$

3.2.4.2 Analysis of Expected Distortion in a Given Partition

We study the expected distortion per utilization of the set S_I. The set S_I has k elements which are $\{q_1^I, q_2^I, \ldots, q_k^I\}$. We derive the average distortion by examining the distortion on each effective element.

The probability of selecting q_1^I in S_I to do a correction can be written as follows:

$$\begin{aligned} P\{\text{Selection of } q_1 | e_{q_1} = x\} &= P\{e_{q_1} = min\{e_{q_1}, e_{q_2}, \ldots, e_{q_k}\}\} \\ &= P\{e_{q_2} \geq e_{q_1}, e_{q_3} \geq e_{q_1}, \ldots, e_{q_k} \geq e_{q_1} | e_{q_1}\} \\ &= P\{e_{q_2} \geq e_{q_1} | e_{q_1}\} P\{e_{q_3} \geq e_{q_1} | e_{q_1}\} P\{e_{q_k} \geq e_{q_1} | e_{q_1}\} \\ &= (1 - F_2(e_{q_1}))(1 - F_3(e_{q_1})) \ldots (1 - F_k(e_{q_1})) \end{aligned} \tag{11}$$

The element q_1 is selected if it is the one with the minimum error in the set. To calculate this probability, we have assumed that the quantization error[3] of different

[3]The phrase quantization error refers to the quantization error after hiding in this section.

elements are independent. This is a reasonable assumption, since the signal to be quantized (DCT transform coefficients) is uncorrelated. The cumulative probability distribution function of each coefficient quantization error is shown by $F_n(\cdot)$.

The probability of selecting q_1 in S_1 can be written as:

$$P\{\text{Selection of } q_1 \text{ in } S_I\} = \int P\{\text{Selection of } q_1 \mid e_{q_1}\} f_1(e_{q_1}) d(e_{q_1}) \quad (12)$$

Using the previous two equations, we can calculate the probability distribution of the quantization error on a coefficient given that it is selected.

$$P\{e_{q_1} | \text{Selection of } q_1\} = \frac{P\{\text{Selection of } q_1 | e_{q_1}\} P\{e_{q_1}\}}{P\{\text{Selection of } q_1 \text{ in } S_I\}} \quad (13)$$

From the last equation, we can get the expected distortion per utilization of an element of S_I.

$$\begin{aligned} D_I = {} & E\{e_{q_1} | \text{Selection of } q_1\} P\{\text{Selection of } q_1 \text{ in } S_I\} + \ldots \\ & \ldots + E\{e_{q_2} | \text{Selection of } q_2\} P\{\text{Selection of } q_2 \text{ in } S_I\} + \ldots \\ & \ldots + E\{e_{q_k} | \text{Selection of } q_k\} P\{\text{Selection of } q_k \text{ in } S_I\} \end{aligned} \quad (14)$$

It should be noted that the only factor needed to determine the expected distortion on usage of S_I is the probability distribution of the coefficient quantization errors.

We present an illustration of this process. The distribution of the quantization error depends only on the distribution of the source, [30, ch. 4.7]. But one important "exception" to this fact is the case of small quantization step-size in comparison with the signal value. For this case, the quantization error can be assumed to be uniformly distributed. For the effective elements of the set, which are the elements with the small quantization step-sizes, this assumption is applicable.

In this illustration, it is assumed that the set S_I has three effective elements, $S_I = \{4, 5, 7\}$ and their quantization error is uniformly distributed as shown at the top part of Figure 17. The bottom part of the figure shows the evaluation of equation

(11) for this case. One can get the probability of using the element "4" for a correction in set S_I as 0.77. For the elements "5" and "7", the same probability is 0.22 and 0.01 respectively. The average distortion for the usage of elements "4", "5" and "7" are $2.84, 2.99$ and 3.6564. From this data, we can calculate the average distortion per usage of S_I as 2.88.

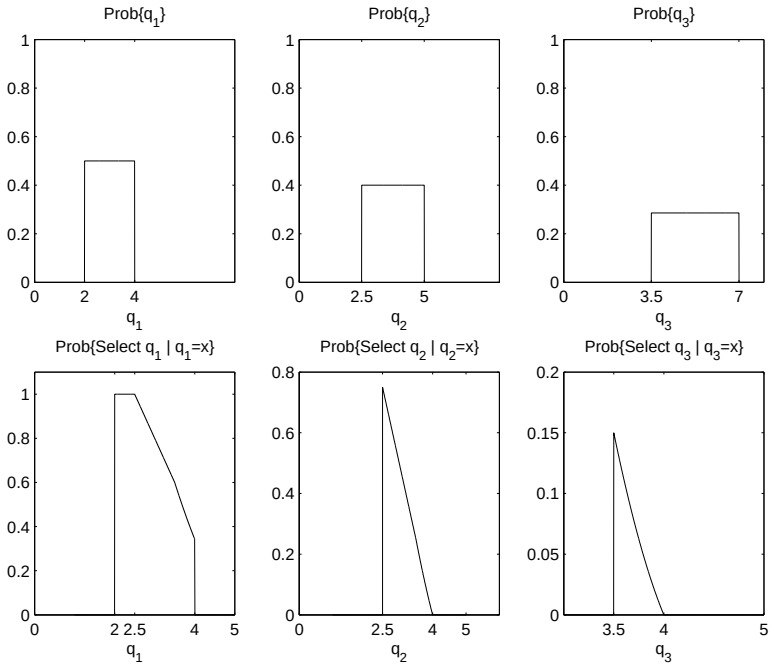


Figure 17: Calculation of Average Distortion in a Partition

3.2.4.3 Arguments on Utilization Factor

We present some arguments on the utilization factor of different partitions. The utilization factor is a difficult system parameter to analyze. In here, we give some informal considerations and examine the consequences of these consideration at the next section. A very low or a very high utilization factor implies an anomaly in the designed competitive system for data hiding. We hope to design a perfect or nearly perfect competitive system for distortion minimization.

The trellis shown in Figure 14 is used to compare the cost of different hiding options. The state of the trellis describes the correction move needed to embed the

desired set of hidden bits.

Assume that an 8 by 8 block is already quantized by the compression device. Let's say that after this operation, the JPEG codeword lies in the partition $(1, 1)$, signaling the hidden bits 1 and 1. If data to be embedded is $(1, 1)$, we do not need to change any coefficient of the quantized block. If data to be embedded is something other than $(1, 1)$; then at least one coefficient should be modified to correct the hidden bits signaled.

The $(0, 0)$ state of the trellis represents the initial state at which no correction action is taken. The final state represents the target state, the state of the required action. If the intended hidden bits are (h_A, h_B) and if the partition of the quantized word is (s_A, s_B), i.e. $(1, 1)$ in the above case, then the target state of the trellis in Figure 14 is $(b_A \oplus s_A, b_B \oplus s_B)$ where the addition is in binary.

If the embedded bits are random and have the same probability of being 1 or 0 (Binomial with $p = q = 1/2$) then the distribution of the target states is uniform, irrespective of the distribution of s_A and s_B. That is, all states at the right-hand side of the trellis is equally utilized.

Secondly, the signaled hidden bits before the data hiding operation, (s_A, s_B), are the binary summation of a group of quantized coefficients, see (3). As in the previous paragraph, if one of the components of the binary addition has the uniform binomial distribution, the sum has the same distribution. For the coefficients with high quantization step sizes (high frequency coefficients), the quantized value is most likely to be zero at their modulo 2 reduction (especially for poor detail, heavily compressed images). Therefore for these coefficients the uniform Binomial distribution assumption is not appropriate. But the assumption is more applicable to the low frequency coefficients. Additionally, we know from information theory that each additional randomization stage increases the entropy of the variable. For example, summing up 10 numbers in binary form is equivalent to starting from a random state of 1 or 0

and then going through 10 stages of binary symmetric channels with different cross-over probabilities. It can be shown that at each stage, the entropy of the variable increases, [17, Ch.2 Problem 31]. After a sufficient number of stages, it is natural to expect to be close to the maximum entropy point[4]. The maximum entropy is the uniform Binomial distribution. As a result, we can be fairly certain that the signaled hidden bits before the hiding operation, (s_A, s_B), is also uniformly distributed.

Due to the described symmetry at the input and the output of the correction system, we may think that the utilization of each set, i.e. usage of the different branches of the trellis, should be the same. Unfortunately, the usage of a branch in the trellis depends on the distortion cost of all branches. This is the main reason of the difficulties at estimating the utilization factor.

Instead of trying to evaluate the utilization of a set in a given strategy, we examine a symmetric solution known to be an equilibrium point due to the discussed symmetry. If it is possible to have identical partitions, i.e. $S_I^* = S_{II}^* = S_{III}^*$, the utilization of these partitions should be exactly the same (under the view of symmetric input/ouput conditions). Unfortunately, the quantization step sizes of a quantization matrix do not always appear in triplets. But it is important to note that if this is the case, the identical distribution of quantization step sizes to the partitions provides an equilibrium point strategy.

In this analysis, we target nearly symmetric distribution when the perfect symmetry can not be achieved. For a nearly symmetric distribution, we expect the individual set distortions and the individual utilization factors to be very similar.

[4]This is the reason, we are assigning not only the effective elements, but also the uneffective elements to the partitions.

3.2.4.4 Optimal Partitioning Strategies

The overall distortion per embedding is

$$D = D_I U_I + D_{II} U_{II} + D_{III} U_{III}. \tag{15}$$

In the previous sections, we have given a method to calculate the value of the expected distortion per partition utilization. We could not present an exact analysis for the utilization factor, but from the discussions given, we expect the set utilization not to vary very much, if $D_I \cong D_{II} \cong D_{III}$.

We propose a utilization rule which captures the behaviour of the utilization factor around a symmetric or nearly symmetric $D_I \cong D_{II} \cong D_{III}$ operating point.

Utilization of a set depends on the average distortion per usage of that set and the average distortion of its competitors. For example if one of the partitions is very badly designed and if the average distortion on that partition is infinity, the encoder is expected not to select that partition at all.

We propose the following rule as the utilization factor around the symmetric operating points:

$$U_I = \frac{1}{\frac{D_I}{D_{II}} + \frac{D_I}{D_{III}} + 1} \tag{16}$$

With the adoption of this rule, we can rewrite the overall distortion equation as:

$$D = \frac{3 D_I D_{II} D_{III}}{D_I D_{II} + D_I D_{III} + D_{II} D_{III}} \tag{17}$$

The utilization rule proposed is based on the symmetry arguments discussed before (reduces to 1/3 for $D_I = D_{II} = D_{III}$). It is inversely proportional to its own distortion, that is sets causing more distortion are utilized less ($D_I \to \infty$ then $U_I \to 0$). Furthermore the final cost function is also symmetric in parameters D_I, D_{II} and D_{III}.

Our aim in proposing this rule is to catch the behaviour of the utilization factor around the symmetric operating point which is known to be an equilibrium strategy.

We have run a computer search to determine the optimum strategy for the 2 bit hiding case using the quantization table entries given in (5). The minimum cost strategy according to this analysis is:

$$\begin{aligned} S_I &= \{4, 6, 6, \text{uneffective elements}\} \\ S_{II} &= \{4, 6, 6, \text{uneffective elements}\} \\ S_{III} &= \{5, 5, 5, 6, 6, 7, 7, 8, 9, 9, \text{uneffective elements}\} \end{aligned} \tag{18}$$

The best strategy turns out to be identical for the sets S_I and S_{II}. The third strategy can not be chosen exactly as the same as S_I or S_{II}, since there are only two 4's in (5). According to the result of this search, the best strategy is placing all other effective elements to the set S_{III}. The distortion of per usage of partitions is $D_I = D_{II} = 2.94$ and $D_{III} = 3.06$ which is reasonably close to each other. This completes the analysis on partitioning.

3.3 Search For Best Partitioning Strategy

In this section, the previous analysis on the two bits per block hiding at the JPEG compression level of 80 is generalized to the arbitrary compression-embedding bitrates. An exhaustive search is made to find the best partitioning strategy at different compression-hiding bitrate pairs. Since the determination and the storage of the optimum partitioning strategies for all possible bitrates is not feasible, a general ad-hoc partitioning strategy is The performance of the ad-hoc strategy is compared with the optimum strategies.

In the previous section, the analysis of the partitioning is given for JPEG quality factor of 80. The default quantization table at this quality factor is:

$$\text{Default JPEG Quantization Matrix at Quality Factor 80\%} = \begin{bmatrix} 6 & 4 & 4 & 6 & 10 & 16 & 20 & 24 \\ 5 & 5 & 6 & 8 & 10 & 23 & 24 & 22 \\ 6 & 5 & 6 & 10 & 16 & 23 & 28 & 22 \\ 6 & 7 & 9 & 12 & 20 & 35 & 32 & 25 \\ 7 & 9 & 15 & 22 & 27 & 44 & 41 & 31 \\ 10 & 14 & 22 & 26 & 32 & 42 & 45 & 37 \\ 20 & 26 & 31 & 35 & 41 & 48 & 48 & 40 \\ 29 & 37 & 38 & 39 & 45 & 40 & 41 & 40 \end{bmatrix} \tag{19}$$

The optimum way of partitioning the block into three sets has been found as:

$$\begin{aligned} S_I &= \{4, 6, 6, \text{uneffective elements}\} \\ S_{II} &= \{4, 6, 6, \text{uneffective elements}\} \\ S_{III} &= \{5, 5, 5, 6, 6, 7, 7, 8, 9, 9, \text{uneffective elements}\} \end{aligned} \tag{20}$$

The phrase "uneffective element" refers to anyone of the elements of the quantization matrix that is not explicitly in one of the sets.

The search operation for the optimum partitions is repeated at the compression levels of QF={10,20,40,80} for the embedding bitrates of 1 to 10 bits per block. It should be noted that the search process is time-consuming due to its combinatorial nature. We have tried to eliminate the redundant comparisons as much as we can by considering the symmetries in the labeling and ordering. In spite of the improved speed, we can not say that the optimum partitions can be determined in a cost-efficient way. An ad-hoc strategy for partitioning is developed because of this difficulty.

We would like to give a listing for the optimum partitions for the QF=10. The results include 1 to 4 bits per block embedding at this quality factor. You can find the complete listing in the Appendix.

```
For JPEG Compression Quality Factor = 10


Quantization_Matrix =


    80    55    50    80   120   200   255   255
    60    60    70    95   130   255   255   255
    70    65    80   120   200   255   255   255
    70    85   110   145   255   255   255   255
    90   110   185   255   255   255   255   255
   120   175   255   255   255   255   255   255
   245   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255


 Embedding Rate : Distortion Value : Partition
 1 : Distortion = 32.63 :
     Partition = { [50  55  60  60  65  70  70  70  80  80 80  85  90  95] }
 2 : Distortion = 35.66 :
     Partition = { [50  65  70  70  80], [55  60  60  70  80  80  85  90  95] }
```

```
3 : Distortion = 37.82 :
    Partition = { [50  80], [55   65   70   80   80  110  110],
                  [60  60  70  70  85  90  95] }
4 : Distortion = 39.59 :
    Partition = { [50], [55  70  85], [60  65  80  80  90],
                  [60   70   70   80   95  110  110] }
```

The 1 bit per block embedding case includes all available step sizes. The 1 bit case is the only trivial partitioning strategy. The 2 bit per block case has 5 elements in one partition and 9 elements in other. The distribution of the step sizes to the partitions does not have a visible pattern. It should be noted that the two smallest step sizes (50 and 55) appear in different partitions. The other step-sizes are distributed in such a way that the distortion caused by each partition is almost the same. Similar comments can be extended for the other cases. In Figure 18, the expected distortion per hidden bit is given

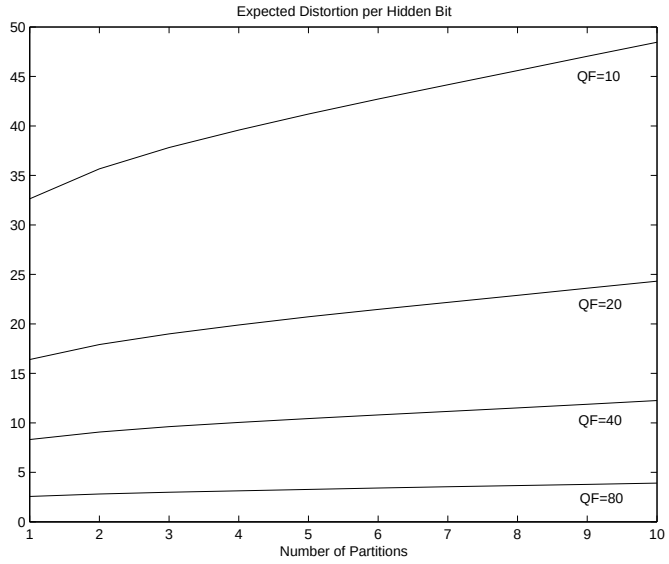


Figure 18: Expected Distortion Per Hidden Bit For Optimal Partitions

3.3.1 An Ad-Hoc Partitioning Strategy

Searching the best strategy at every compression-embedding level is a difficult task. The task is even more difficult for the video systems using multiple quantizers for compression. We examine a simple ad-hoc strategy that is applicable to all bitrates in this section. This strategy can partition any JPEG quantization table into any subsets.

The strategy for k partitions can be described as follows:

1. Sort the quantization step-sizes from the smallest to the largest,
2. Make a matrix with k columns and sufficient number of rows to accommodate 64 elements,
3. Fill the matrix with the sorted data rowwise (start filling the first row and then second),
4. Scan the matrix along the diagonals (see Figure 19),
5. Copy the elements on k diagonals to partitions.

The ad-hoc partitioning strategy for QF=80 for three partitions is:

$$\begin{aligned} S_I &= \{4, 5, 6, 6, \text{uneffective elements}\} \\ S_{II} &= \{4, 6, 6, 6, \text{uneffective elements}\} \\ S_{III} &= \{5, 5, 6, 7, 7, 8, 9, 9, \text{uneffective elements}\} \end{aligned} \tag{21}$$

The strategy is different from the optimal one given in mrefoptimaloneschap3. The expected distortion for the ad-hoc partitions are $D_I = 2.85$, $D_{II} = 2.93$ and $D_{III} = 3.24$. The same values for the optimal sets are $D_I = D_{II} = 2.94$ and $D_{III} = 3.06$. The overall expected distortion per bit is 2.97 for the optimal sets, and 3.00 for the ad-hoc set.

As noted before, the assignment of the step-sizes to the partitions is a resource allocation or a logistics problem. The contractors of the logistics problem are the partitions. And the resources that are being assigned to the contractors, are the quantization step sizes. Smaller step sizes are more valuable than the larger ones. We seek to distribute the resources to the contractors in a fair way. To keep the system as competitive as possible, we try not to favor one partition to another and assign the resources to the partitions in such a way that the resulting set distortions are almost equal after partitioning. If the cost per usage of partitions are almost the same, then the partitions are expected to be utilized almost equally which is in agreement with the analysis in the previous section. In short, to increase the efficiency of the whole system the resources are tried to be assigned in an egalitarian way. Since the optimum way is too complicated to determine and an ad-hoc strategy is developed.

The ad-hoc strategy resembles the draft system of the NBA league. In the NBA system, the least successful team of the season is allowed to make the first selection in the pool of draftees. The second least successful team makes the second choice and the champion of the season has the last choice. Teams can not select another player until other teams are finished with their choices in that round. The proposed ad-hoc system is analogous to this strategy. The goals of both systems are the same which is to keep the competitiveness as high as possible.

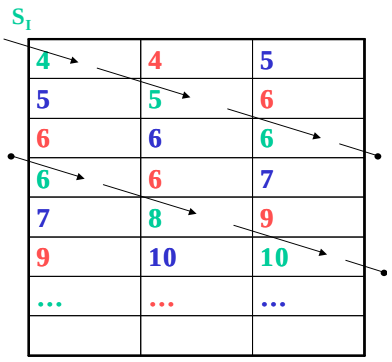


Figure 19: The Ad-hoc Method For Three Partitions

In the Figure 20, the ad-hoc strategy is compared with the optimal strategy. It is clear that the ad-hoc rule closely follows the optimal rule. We give the ad-hoc partitioning result for the case whose optimal sets are given before. The readers are invited to compare these two strategies.

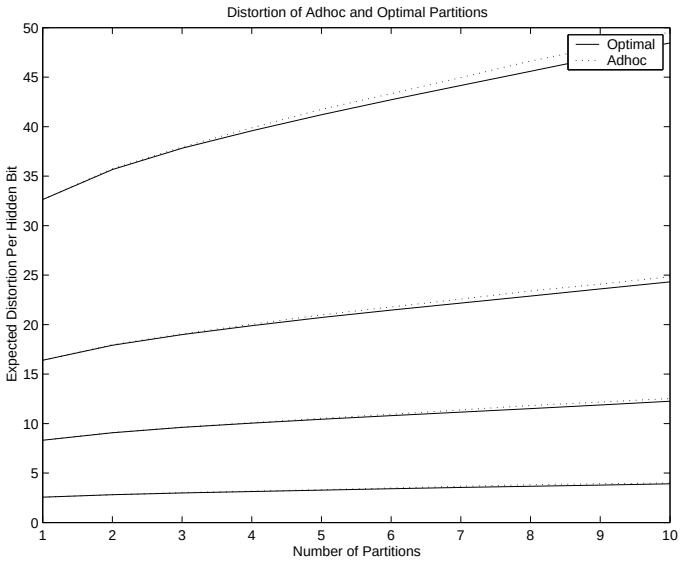


Figure 20: Distortion Comparison of Adhoc and Optimal Partitions

```
For JPEG Compression Quality Factor = 10


Quantization_Matrix =


    80     55     50     80    120    200    255    255
    60     60     70     95    130    255    255    255
    70     65     80    120    200    255    255    255
    70     85    110    145    255    255    255    255
    90    110    185    255    255    255    255    255
   120    175    255    255    255    255    255    255
   245    255    255    255    255    255    255    255
   255    255    255    255    255    255    255    255
```

```
Embedding Rate : Distortion Value : Partition
1 : Distortion = 32.63 :
    Partition = { [50  55  60  60  65  70  70  70  80  80  80  85  90  95] }
2 : Distortion = 35.75 :
    Partition = { [50  60  65  70  80  85  90], [55  60  70  70  80  80  95] }
3 : Distortion = 37.94 :
    Partition = { [60  60  70  85  90], [55  70  70  80], [50  65  80  80  95] }
4 : Distortion = 39.88 :
    Partition = { [50  70  80], [55  70  85  90], [60  70  80  95],
                  [60   65   80  110] }
```

3.4 JND Guided Data Embedding

The performance of the hiding algorithm whose optimum parameters are analyzed in the previous sections is improved with a selective embedding process which is guided with a human visual system model. The described improvement lets us to embed more bits at the distortion insensitive blocks (busy or low contrast blocks) and fewer bits to the distortion sensitive ones.

The basic idea is making effective usage of the local properties of a given image. The darker and busy blocks of an image have more distortion tolerance than the smooth blocks. If we examine the Lena image, the distortion on the rim of the mirror, or on the fur of the hat is less perceptible than the distortion on the shoulder. The algorithm as described in the previous sections embeds the same number of bits in all blocks. The improvement described in this section allows us to do a selective embedding depending on the image features.

To determine the local features, Watson's human visual system model is implemented to estimate the Just-Noticeable-Distortion levels of DCT coefficients, [50]. The JND estimation is based on the contrast and the component masking. The contrast masking allows us to distinguish the low contrast blocks. The component masking allows us to determine the busy blocks.

To accomplish the input adaptive embedding without sending any information on the content (otherwise the blindness requirement would be violated), a randomization stage is proposed. At the first step, the JND values of DCT coefficients are calculated. At the next step, the block coefficients are randomly shuffled. At the last step, the options of embedding are evaluated using the JND weighted errors on the shuffled coefficients. The shuffling is described further in this section.

A simple explanation can be given as follows: If all DCT coefficients in a block have infinite JND values, all coefficients of this block can be used for hiding. These coefficients create no perceptible distortion until their value is changed by their JND

value which is the infinity in this case. The uniform embedding algorithm treats every block identically. The HVS improved algorithm uses shuffling to distribute the 'good' coefficients into the blocks. With the JND weighting, these good coefficients are always selected for the modification where ever they reside. The shuffling pattern is needed for decoding. In order not to violate the blindness requirement, we propose to include a random number generator seed in the design. The same seed is used at the encoder and decoder to generate the shuffling pattern.

3.4.1 Watson's Human Visual System Model

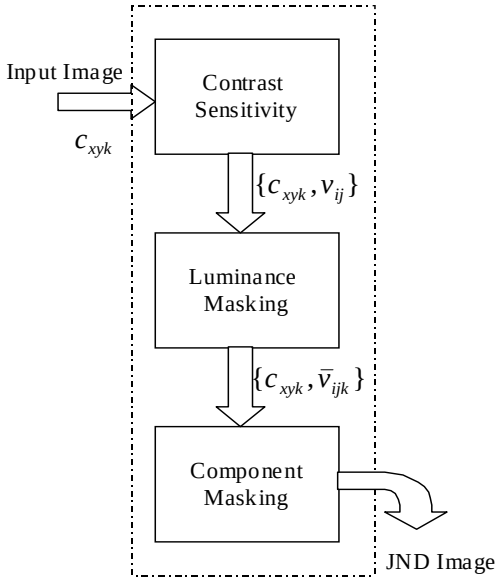
Watson's human visual system model is used to determine the perceptual significance of the DCT coefficients, [50]. The perceptual significance of the transform coefficients in a block are the updated versions of the image-indepedendent frequency sensitivity values of the DCT basis functions. Watson's model updates the image-independent levels by taking the local properties of the image into account. As can be seen from Figure 21, Watson's model includes the effect of the contrast and component masking of the human visual system. The updated levels are labeled as Just Noticeable Difference (JND) levels of the coefficients. The JND level of a component can be interpreted as the maximum amount of the algebraic modification which is not perceptible.

The pictures in Figure 22 show the original image and its JND levels. The low contrast regions of the image such as the rim of the mirror and the busy regions such as the fur on the hat have high JND values. The smooth regions such as the shoulder of Lena and her reflection on the mirror have low JND levels.

3.4.2 Embedding with JND Weighting and Shuffling

After JND values are calculated, the algebraic error due to data hiding modifications is weighted by the JND value of the modified coefficient. The goal of this operation is to include the information on the block and its neighborhood in the embedding decision. The resulting error metric is called JND weighted MSE.

Watson's HVS Model



Contrast Sensitivity: Sensitivity of each DCT basis function is measured by Ahumada [2] and adopted by JPEG committee at the design of the default quantization table.

Luminance Masking: Suggested by Watson as:

$$\bar{v}_{ijk} = v_{ij}(c_{00k} / \bar{c}_{00})^{-a_T}$$

V_{ij}= Contrast sensitivity of (ij)th DCT basis function

C_{ijk} = DCT {block #k}

a_T= Luminance masking parameter

Component Masking: Suggested by Watson as:

$$S_{ijk} = \bar{v}_{ijk}.\max[1, |c_{ijk} / \bar{v}_{ijk}|^{w_{ij}}]$$

w_{ij}= Component masking parameter

S_{ijk} : JND Image

References: [1] A. B. Watson, "DCT quantization matrices visually optimized for individual images," Proc. SPIE, 1913:202-16, 1993.
[2] A. J. Ahumada Jr., A. B. Watson, & H. A. Peterson, "A visual detection model for DCT quantization," AIAA Computing in Aerospace, pp. 314-318, San Diego, CA, 1993.

Figure 21: Watson's Human Visual System Model

Shuffling: The idea of shuffling has appeared in the literature before. You can look at [1, 55] for different usages of shuffling. Dr. Faisal Al-Turki uses the idea of shuffling to increase the security of the watermarking system. Dr. Wu uses it for the payload equalization.

We use shuffling to mix the good coefficients (the coefficients of the distortion insensitive blocks) with the bad ones. Prior to shuffling operation, the 'good' coefficients are concentrated in the good blocks (low contrast, busy blocks) and the bad ones are localized elsewhere. After the application of shuffling, the good, the bad and the ordinary coefficients are expected to be distributed uniformly in the image. The increased diversity of the coefficients in blocks increases the performance of the system. We would like to note that the previously described uniform embedding

(a) Lena Image

(b) JND Levels

Figure 22: JND levels of Lena Image, lighter colors show higher JND values.

algorithm is tailor made for the uniformly distributed coefficients.

We propose the following shuffling method. It is called the channel shuffling:

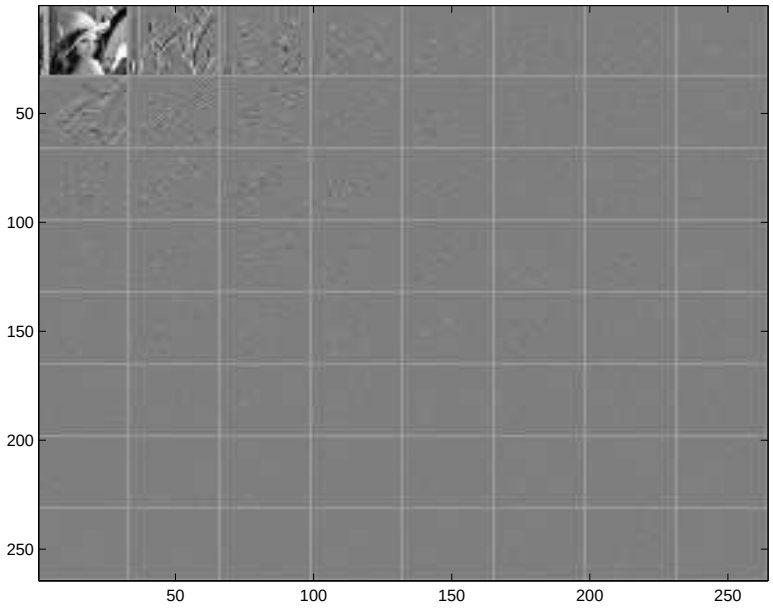
- Gather the kth DCT coefficients from every block (0th DCT coefficients (DC), 1st AC coefficients etc.)
- Randomly permute the collected coefficients among themselves
- Distribute randomized coefficients to their corresponding blocks

The channel shuffling permutes kth DCT coefficients of the blocks among themselves. In Figure 23, we illustrate the shuffling. The block looking like a miniature Lena image is the collection of DC coefficients. The one on its right is the collection of the first DCT coefficients. Some edge structure is visible in this one. Other DCT coefficients are collected similarly. The shuffling occurs among the coefficients with the same DCT index. And it results in a picture as in Figure 23.

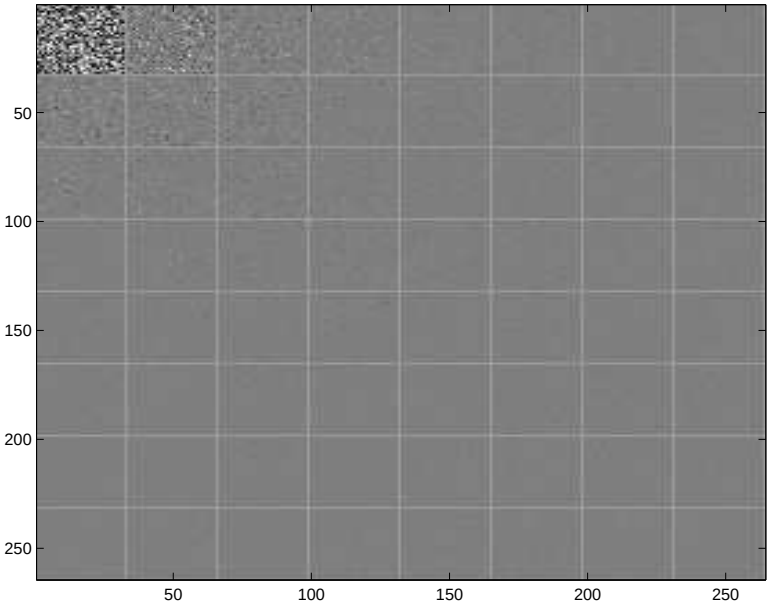
Combining Shuffling and JND: By combining shuffling and JND ideas, we reach the desired goal. The good blocks with lots of high JND coefficients can be used more frequently with this improvement.

The Figure 24 illustrate the results. At this simulation four bits per block is embedded to Lena image at quality factor of 60. The digits represent the number of modified coefficients in a block. It can be noticed that this number is always less than 4 in the top picture.

For the JND based approach, the number of modified coefficients in a block jumps to higher numbers than 4 where ever it is advantageous. It can be seen that the smooth regions of the image have less modified coefficients. We can say that the approach of shuffling with JND weighting keeps the smooth regions smooth by embedding more bits into the noisy or low contrast regions. The Figure 25 illustrates the usage of different metrics.



(a) DCT Channels



(b) DCT Channels After Shuffling

Figure 23: DCT Channels Before and After Shuffling

(a) PSNR

(b) wPSNR+ Shuffling

Figure 24: The Number of Modified Coefficients in a Block with PSNR and JND weighted PSNR metric

PSNR Optimized

PSL_1E Optimized

wPSNR Optimized

$wPSL_1E$ Optimized

Figure 25: Data Hiding Results With Different Objective Metrics

3.5 Experiments On The Method

This section includes a report of the conducted subjective quality tests and the results of some functional tests on the method. For these experiments we have used a Matlab implementation of the method which is described below.

Matlab Implementation: The interface of the program is shown in Figure 26. The input and output parameters are as follows:

Input Parameters:

- A bitmap image
- JPEG compression factor (quality factor)
- Embedding Rate
- Hidden Data

Output: JPEG image

Distortion Metrics: These are the distortion metrics that can be used for embedding:

- PSNR
- $\mathcal{L}_1$
- PSNR with JND weights
- $\mathcal{L}_\infty$ with weights

JND weights: These are the parameters for Watson's human visual system model:

- Contrast Masking : Higher the parameter, the more is the masking
- Component Masking : Higher the parameter, the more is the masking

Figure 26: The Interface of the Matlab Program

The implementation allows us to embed the same data using different metrics or different HVS parameters. The embedding-compression rate can also be altered. Program displays before and after hiding images in adjacent panels. The error and bitrate information is displayed below the images. The Figure 26 shows an application on the Lena image at at QF=60 with 10 hidden bits/block embedded using MSE with JND weighting and the partitioning method is the ad-hoc method (This information can be read in the name of the image file).

Subjective Tests: The first test is to determine the zero-perceived distortion hiding capacity. The different stages of this test is taken by 9 to 14 subjects, seated at the same initial viewing conditions. The subjects were allowed to adjust the viewing conditions for their comfort. The test questions are randomized. The placing of the images is also randomized. We have tried to implement a doubly blind test (experiment and subject can not infer any information on the experiment) with the random placement actions. At some cases, we have asked the same twice to check the consistency of the subject.

The second test is on the utility of the JND based approach. The promising approach is compared with the non JND approach. This test is taken by almost 100 subjects located as near as next cubicle and as far as Korea. Since this test is a simple preference test, we established a web site to compare 15 images. The ease of the setup attracted many subject whom we are thankful for their time and effort. The test is done in a similarly doubly blind fashion. Web site visitors (subjects) can use aliases if they do not want to be identified.

3.5.0.1 Test of Zero-Perceived Distortion Hiding Capacity:

The goal is to measure visibility of the embedding distortion at different embedding-hiding conditions. The compression and embedding rate are the parameters of the experiment.

As expected when the images are lightly compressed, the embedding distortion gets perceived at a higher embedding rate compared with more aggressive rates. The threshold of visibility at different compression rates is the goal of the test. To achieve this goal, we followed the two step procedure:

Step 1: Estimate an upper and lower bound for the perceptibility threshold

Step 2: Bisect the interval with additional tests to increase the accuracy

Step 1: A ranking test is used to estimate the bounds. Subjects are asked to order the images from the worst quality to the best. Our goal is to determine the interval in which the subjects can not order the images consistency. The images in this interval are assumed to be perceptually indistinguishable. If all subjects can order the images in the same order that the embedding rate decreases, we can say that the distortion in these images are perceptible to all. If there is no common ground in the ordering, we can deduce that the perceptibility threshold is above the tested embedding rates. You can examine the screenshot of the test in Figure 27.

In Table 4, the results for the Lena image at QF=80 is given. The response "0" means that the subject thinks both images are the same, or denies to make a choice. The responses "1" to "4" mean subject selects the image with 0,3,5 and 8 hidden bits per block (respectively). From the subject scores given in this table we can say that visibility threshold is upper bounded by 5 bits/block at QF=80 for Lena image.

Step 2: The established range is bisected to get an accurate estimate on the zero-distortion embedding bitrate. A comparison test is implemented. The Images with and without hidden are compared. Subject responses are analyzed. We have included to some redundancy in the test to test the consistency of the subjects. A screenshot of this stage is given in Figure 28.

The results for the Lena image at QF=50 is given in Table 5. The response "1" in the table corresponds to the image without hidden bits. "2" corresponds to the image with hidden bits. "0" corresponds the case when the subject makes no preference.

Figure 27: Interface of the First Step of Perceptibility Test

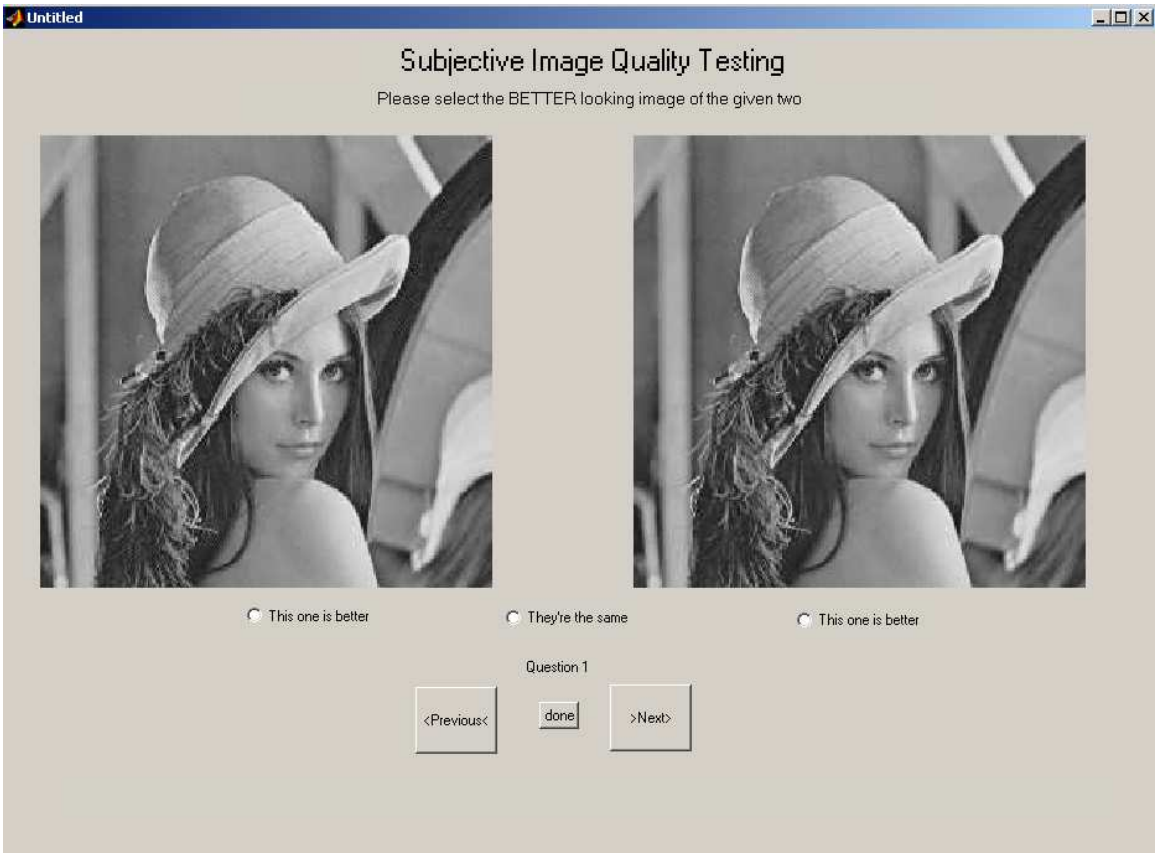


Figure 28: Interface of the Second Step of Perceptibility Test

Table 4: Subjective Quality Test Results for Lena Image at Quality Factor QF=80

	Lena at QF=80			
	Best	...	...	Worst
Subject A	3	2	1	4
Subject B	4	2	3	1
Subject C	1	3	2	4
Subject D	0	0	2	4
Subject E	2	1	3	4
Subject F	1	2	4	3
Subject G	3	2	4	1
Subject H	2	3	1	4
Subject I	1	2	4	3
Subject J	2	1	4	3
Subject K	2	1	4	3

The responses of the inconsistent subjects are changed to no-difference. The consistency check is shown with the $(\cdot)$ operation in the Table.

Conclusion: For Lena image the hiding distortion is not perceived for 1 bpp (or higher) compression rates up to the 5 bits / block. For the Baboon image the same threshold is 7 bits/pixel. For 0.5 bpp compression rate, the threshold 3 bits/block and 4 bits/bloc for Lena and Baboon images respectively.

3.5.1 Test of JND Based Approach

We have examined the effectiveness of JND based approach. The basic idea of shuffling and JND weighting is to embed more bits at locations where they are less visible.

We have established a website for the test. A comparison test is conducted. Twelve pairs of images are presented. Each pair consists of an image with JND based embedding and without JND based approach at the same operating conditions (compression rate and hidden data). The visitors are asked to make a preference between two images. You can see the screenshot of this test in Figure 29.

In Figure 30, you can see the bar graph of the user responses. For low embedding rates, shuffling has little effect. The distortion is not perceptible for low embedding

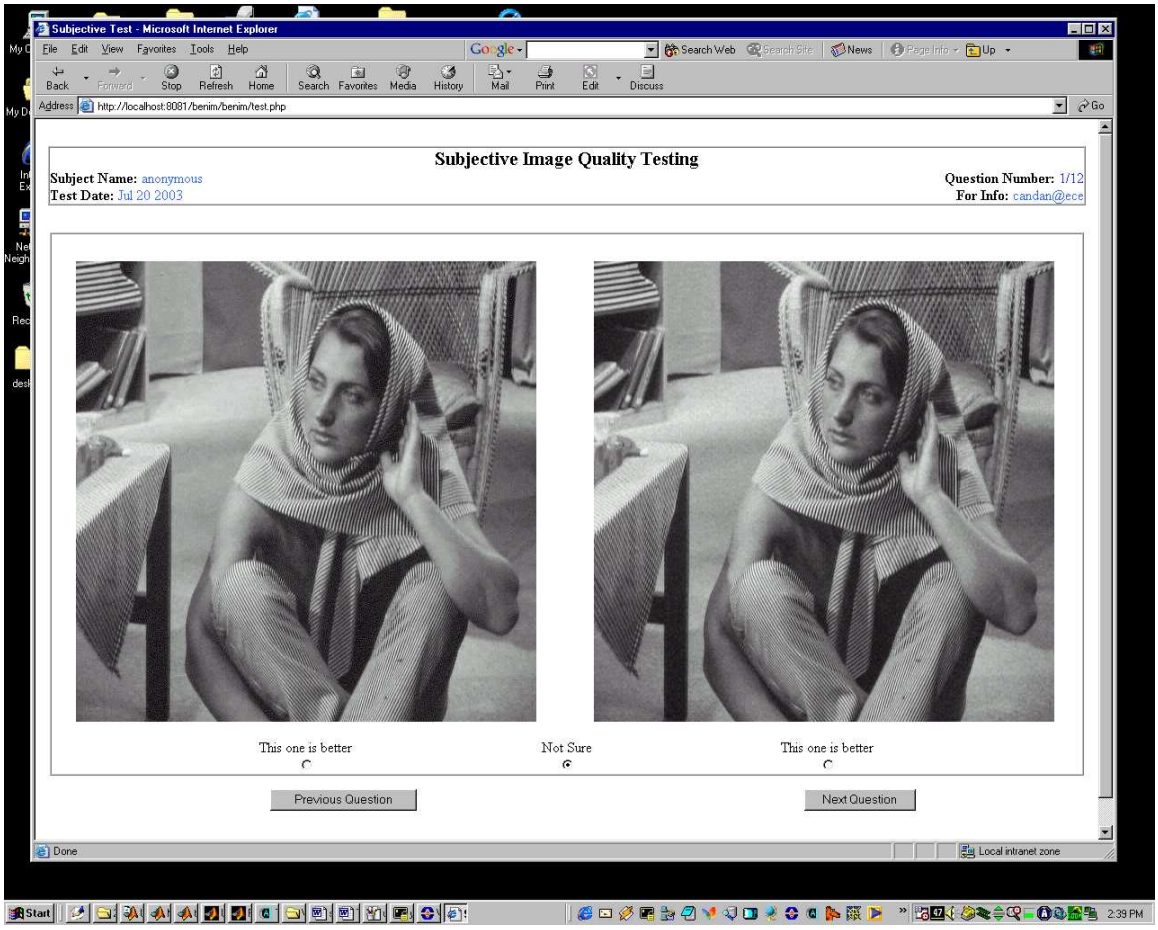


Figure 29: Interface of the Shuffling-JND Weighting Test

Table 5: Subjective Quality Test Results for Lena Image at Quality Factor QF=50

	Embedding Rate (bits/block)					
	1	2	3	4	5	6
Subject A	0	0	(0,0)=0	(0,0)=0	(1,0)=0	1
Subject B	0	0	(1,2)=0	(0,0)=0	(1,1)=1	1
Subject C	0	1	(2,0)=0	(1,1)=1	(1,1)=1	1
Subject D	0	1	(0,0)=0	(1,1)=1	(1,1)=1	1
Subject E	2	0	(2,1)=0	(0,1)=0	(1,1)=1	1
Subject F	0	1	(1,1)=1	(1,2)=0	(1,1)=1	1
Subject G	0	0	(0,2)=0	(0,1)=0	(1,1)=1	1
Subject H	1	1	(2,0)=0	(2,2)=2	(1,1)=1	1
Subject I	0	0	(0,0)=0	(0,0)=0	(1,1)=1	1
Subject J	0	1	(0,0)=0	(1,1)=1	(1,1)=1	1
Result:	**10%**	**50%**	**10%**	**20%**	**90%**	**100%**

rates and the need for shuffling is little. As the embedding rate increases, the subjects are inclined towards to the shuffling option.

3.5.2 Statistical Analysis of Subjective Test

In this section, we present a statistical analysis of the JND based embedding experiment.

A Short Summary of Hypothesis Testing: We would like to quantify the value of the JND based shuffling. To achieve this, the subject responses are mapped to the scores of {-1,0,1}. The score of "1" and "-1" refer the case with JND shuffling and the case with MSE respectively. The score "0" refers to the subject response of "No Difference".

The hypothesis testing is the selection of one of the two complimentary hypotheses. The hypothesis that is being testing is called *null hypothesis* (H_0) and its complement is called *alternative hypothesis* (H_1). A hypothesis is rejected or accepted by the evaluation of the experimental data with a user determined accuracy or degree of certainty. In the terminology of hypothesis testing, the degree of certainty is called *significance level*. If the probability of the hypothesis to be true falls below the significance level,

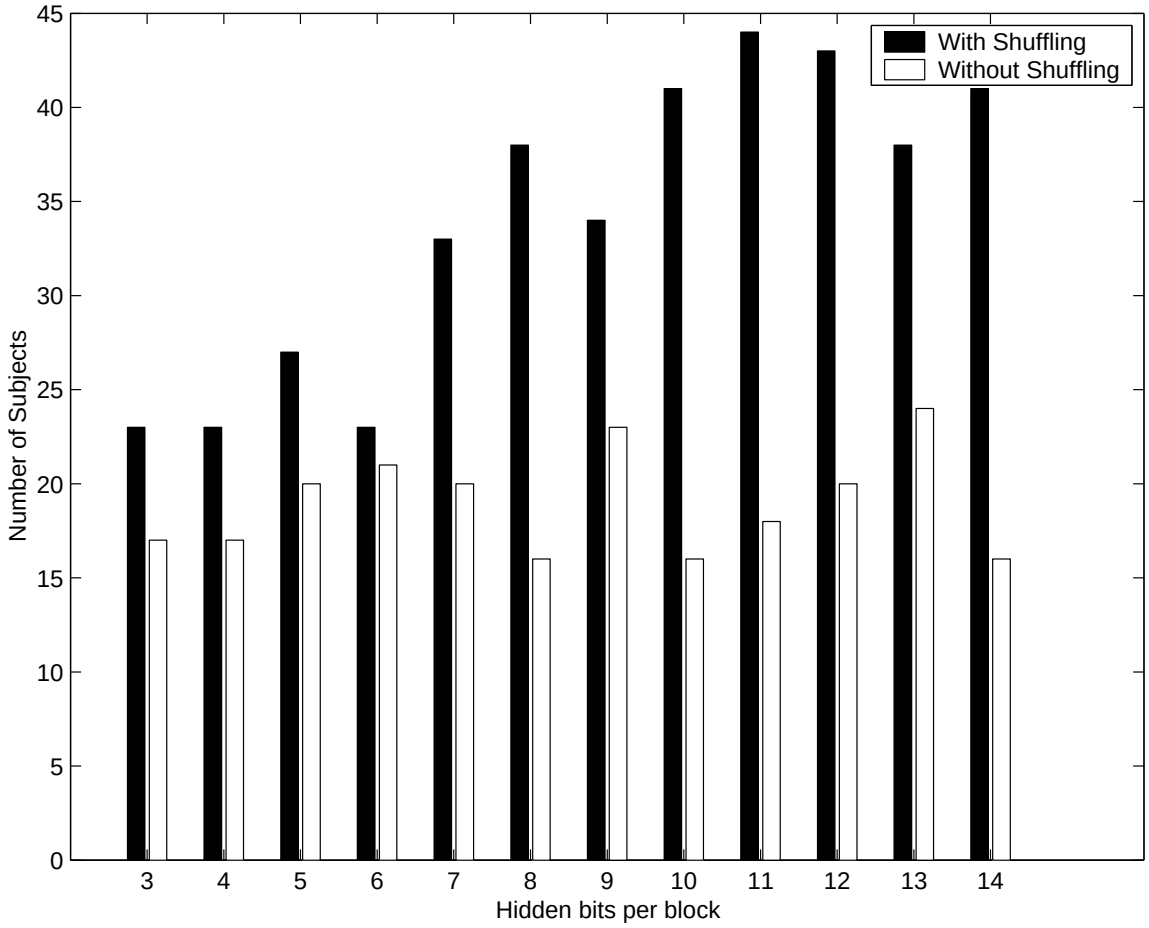


Figure 30: Bar Graph of Subject Preference With and Without JND Weighting

the hypothesis is dropped. In other words, the probability of incorrectly rejecting the null hypothesis when it is actually true is always less than the significance level. A second word of terminology that we need for the analysis is the *confidence interval*. The confidence interval is the range of values for the hypothesis to be held as correct (at a significance level). For example, if the null hypothesis is the temperature of Atlanta in December being 50'F, the range of temperature values around 50'F that cannot be rejected as incorrect form the confidence interval.

For our test, we construct the null hypothesis as the two options of embedding having no difference. Any deviation from the null hypothesis is an interesting event. The null hypothesis is:

$$H_0 : \mu = 0 \tag{22}$$

where μ is mean of the user preference distribution. The alternative hypothesis is:

$$H_1 : \mu \neq 0 \tag{23}$$

The significance level is set as 0.05. The confidence interval is therefore 95% by its definition of $1 - \nu$, where ν is the significance level.

Two sided t-test is used to test the hypothesis. The two sided t-test does not require any statistical information such as the variance or the probability distribution of the source. The T-test uses the sample set to estimate the variance of the random variable and evaluates the hypothesis and the deviation range from the estimated mean and variance. The operation of T-test is based on the Tchebycheff's inequality:

$$P\{|\overline{x} - \mu| > \epsilon\} \leq \frac{\sigma^2}{\epsilon^2} \tag{24}$$

In the above relation, the deviation from the mean is shown by ϵ. $\overline{x}$ refers to the estimated mean. The parameters μ and σ^2 are the true mean and variance of the source. The Z-test that we have not used, uses the variance of the source to evaluate the hypothesis. The T-test uses an estimate of the variance for the same

purpose. It is known that the estimated variance converges to the true variance with $1/n^2$ rate, see [38] (n is the number samples in the set). Therefore we may expect the estimate and actual values to be close for large sample sets. Since the sample size of our experiment is 98 subjects, we are pretty comfortable with the variance and mean estimates. Readers may refer to an elementary statistics book for the details on the T-test, [32]. An implementation of the T-test is available in the MATLAB Statistics Toolbox.

In Figure 31, we show the mean of the sample set with the 95% confidence interval at different embedding bitrates. The input data of the T-test has been plotted as a bar-chart in Figure 30. As previously noted, the shuffling and JND based method clearly outperforms the MSE based method at high embedding bitrates. At low embedding rates, the performance of both methods are similar; but the JND-shuffling method does not underperform at any bitrate.

3.6 Functional Tests

We present some experiment to investigate the limitations of the designed system.

3.6.1 Comparison With The Spread Spectrum Technique

We present a comparison of the described algorithm with the spread spectrum method, [18].

The spread spectrum method embeds hidden data into the DCT coefficients, but the output image is stored in the pixel domain with the bitmap format. In order to compare the distortion levels of two systems, the output of the spread spectrum method is compressed to the operating JPEG compression level of the minimum distortion method. The post-hiding JPEG compression acts as an attack on the spread spectrum method. The spread spectrum method is known to be robust to the JPEG attacks. In this experiment, we have chosen light compression levels on purpose in order not to cause any compression related complications.

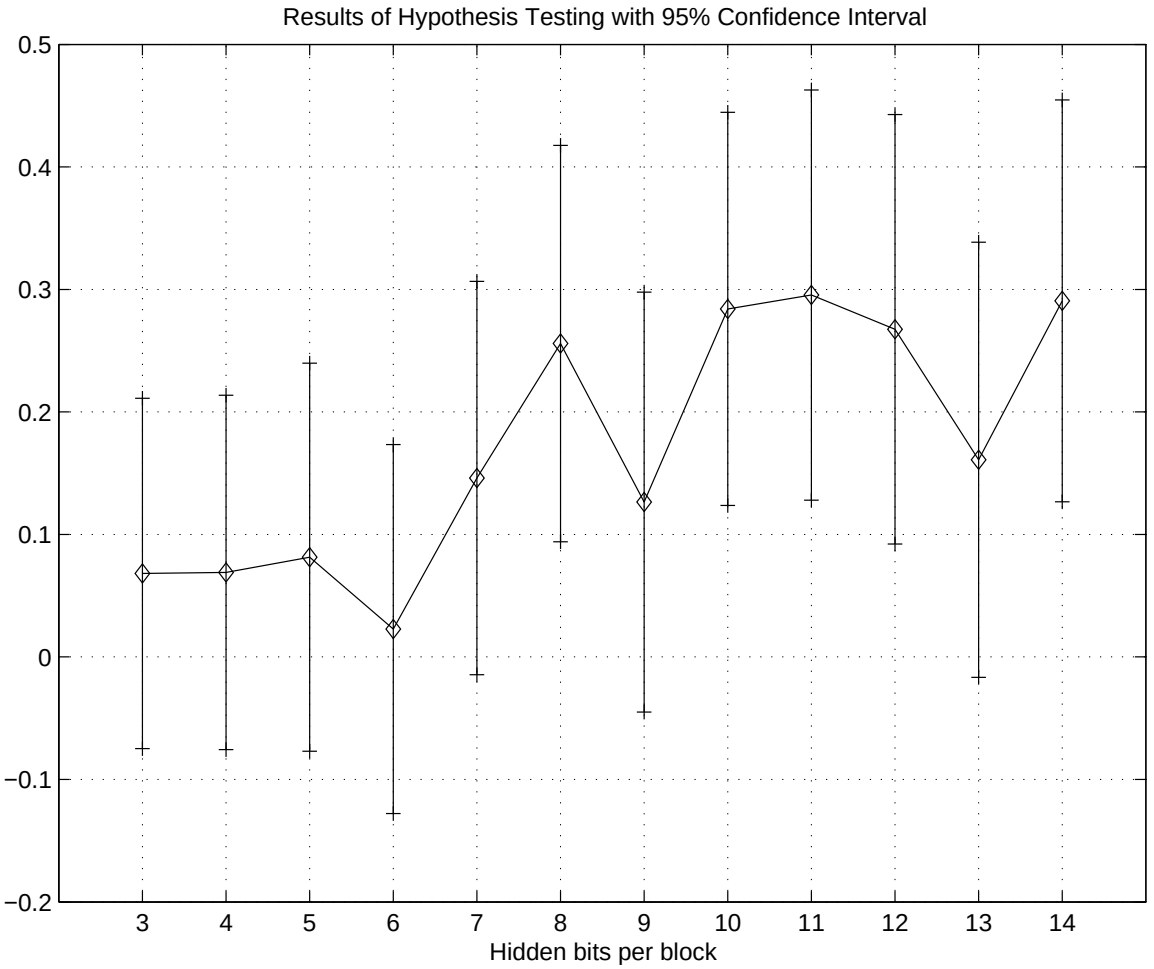


Figure 31: Results of Hypothesis Testing. The sample mean is labeled with the diamond symbol. The interval of 95% confidence around the mean value is shown. The skew towards the positive values illustrate the preference of the JND-based shuffling.

.

The hiding algorithms for robust applications is built on different fundamentals. The watermark energy plays a critical role in the additive robust methods. The distance between the codewords plays a similar role in the quantization based methods. The minimal distortion method can be interpreted as a quantization based method with the least possible minimum distance of 1. Due to the differences in objectives, we expect our algorithm to perform significantly better in terms of distortion performance and significantly poorer in terms of robustness. We have conducted this experiment to illustrate the performance gap between our algorithm and a well known technique when the other technique has to be used at a minimal distortion application.

We have tested the distortion performance on two different compression levels. The first compression level is the default level of 75. The second one is a more aggressive quantization level of 50. We have adjusted the watermark strength that at least 80 percent of the embedded data can be extracted correctly after the post-hiding JPEG compression attack. This leads to the watermark strength of 0.1 (which is the default level recommended by Cox et. al) and 0.2 for the more aggressive case.

The Tables 6 and 7 present the results of this comparison. The PSNR values of the proposed method is significantly higher than the spread spectrum method at all hiding rates. We would like to repeat that the spread spectrum method like any other robust method is not designed for the distortion minimization. The percentage of the correctly decoded bits for both methods is also reported. The minimum distortion method can extract all hidden bits correctly, since its operation domain is the JPEG domain. The percentage of the correctly decoded bits of the spread spectrum method monotonically decreases at a given watermark strength, as more data is embedded.

3.6.2 Effect of Data Hiding on File Length

At this section, we examine the effect of data hiding on file length. JPEG compression is done to reduce the size of the image. Data hiding operation introduces redundancy

Table 6: Comparison of The Designed Data Hiding Method with the Spread Spectrum Method. The test image is the 256 x 256 Lena Image at the quality factor of 75.

Hiding Rate (bit/block)	PSNR (proposed)	PSNR (SS)	Correct Decoding % (proposed)	Correct Decoding % (SS)
1	34.53	31.85	100 %	95.41 %
2	34.52	31.71	100 %	92.68 %
3	34.50	31.68	100 %	90.82 %
4	34.48	31.66	100 %	88.75 %
5	34.45	31.66	100 %	86.88 %
6	34.42	31.65	100 %	84.77 %
7	34.39	31.62	100 %	82.52 %
8	34.34	31.60	100 %	82.09 %
9	34.30	31.60	100 %	80.5 %
10	34.25	31.59	100 %	78.88 %

into the image which may increase its length. At this section, we look at the extension of file length after data hiding.

We experiment on two images. Both images are of size 512 by 512. We have embedded randomly generated data to these images and checked the size of the JPEG file after hiding operation. We have repeated the same experiment at four different compression levels, at different embedding rates.

The results for Lena image is shown in Figure 32. For high quality images, the file size increase is significantly less than the information embedded. For low quality images the size increase is comparable with the information embedded. The bitrates of the images is given at the figure caption.

The results for Barbara image is shown in Figure 33. Barbara image retains more redundancy after compression than Lena image. In other words, the resultant bitrate of Barbara image is higher than the bitrate of Lena image when compressed at the same quality factor (please see the figure captions for the bitrate comparison). Since Barbara image contains more redundancy, it should be easier to embed information into it. It is interesting to note that for the high bitrate Barbara images (QF=80)

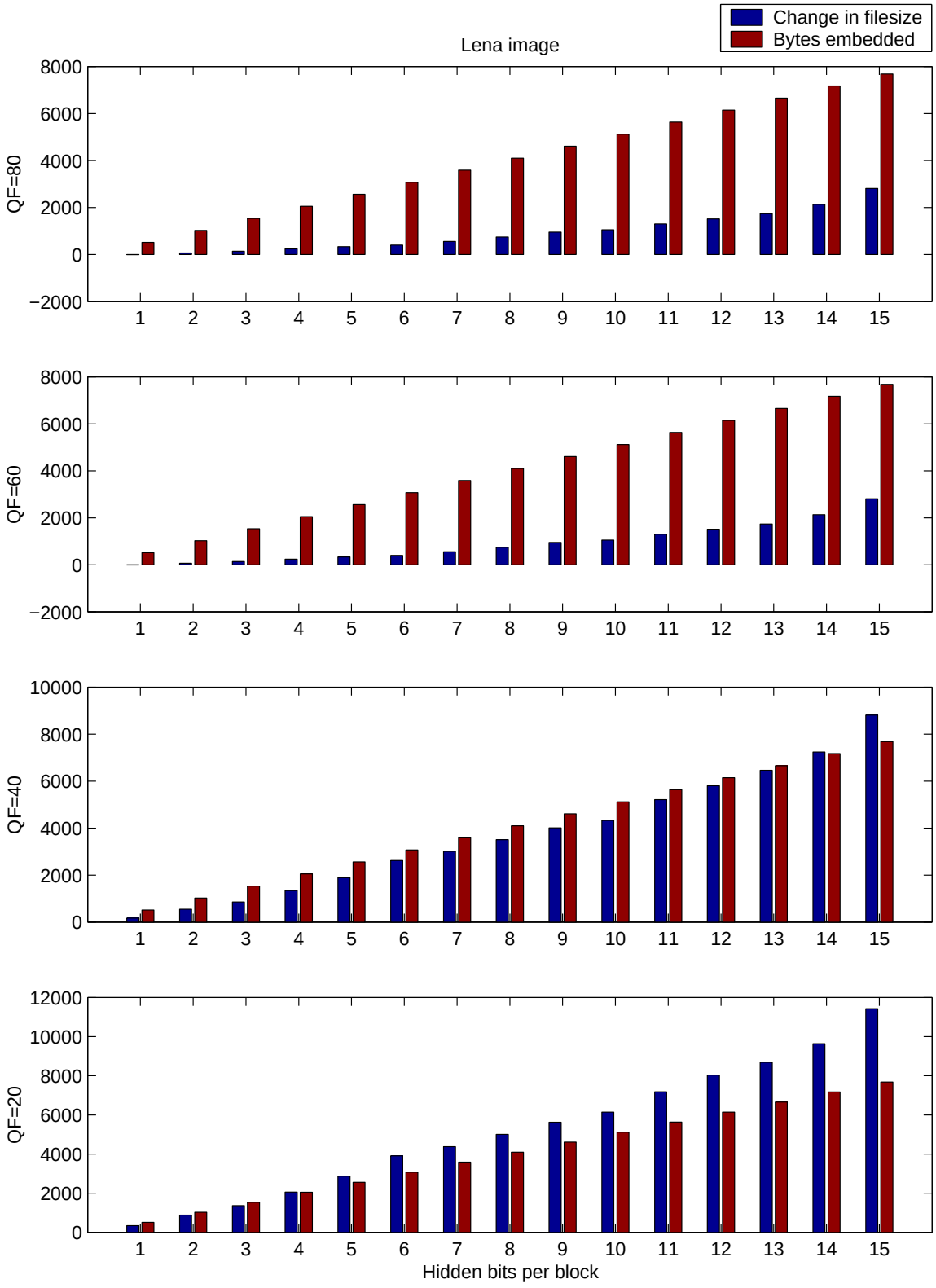


Figure 32: Change in Filesize for Lena Image. The bitrates of the image from highest quality to the lowest are {1.15, 0.73 , 0.54 , 0.36 } bpp.

Table 7: Comparison of The Designed Data Hiding Method with the Spread Spectrum Method. The test image is the 256 x 256 Lena Image at the quality factor of 50.

Hiding Rate (bit/block)	PSNR (proposed)	PSNR (SS)	Correct Decoding % (proposed)	Correct Decoding % (SS)
1	31.96	27.35	100%	94.92%
2	31.94	27.19	100%	92.87%
3	31.90	27.11	100%	90.95%
4	31.84	27.11	100%	88.13%
5	31.77	27.10	100%	86.84%
6	31.69	27.07	100%	85.22%
7	31.60	27.05	100%	83.87%
8	31.51	27.02	100%	82.47%
9	31.42	27.01	100%	80.88%
10	31.32	27.03	100%	78.81%

hiding operation causes a reduction of filesize. This case is interesting since the transmission of the hidden information in Barbara image at this compression rate is totally free of cost, since the after hiding has no perceptible distortion and there is no additional bandwidth cost. The reduction on the file size can be explained by the longer run-length after hiding.

3.6.3 Effect Of Image Size on Hiding Efficiency

We examine the embedding efficiency at different image sizes. The goal of this experiment is to investigate the effect of image size on the data hiding algorithm. For this experiment, we have selected Lena, Fishing Boat and Peppers images and resized them to the dimensions of 128×128, 256×256 and 512×512 using bicubic interpolation. One of these sample images is shown in Figure 34.

The medium size image is compressed at 1 bpp and its PSNR value is recorded. The higher and lower size images are compressed such that the PSNR distortion matches the recorded PSNR. We have visually verified that all images have no systematic artifacts such as blocky artifacts, mosquito noise, stair-case effect. Hence

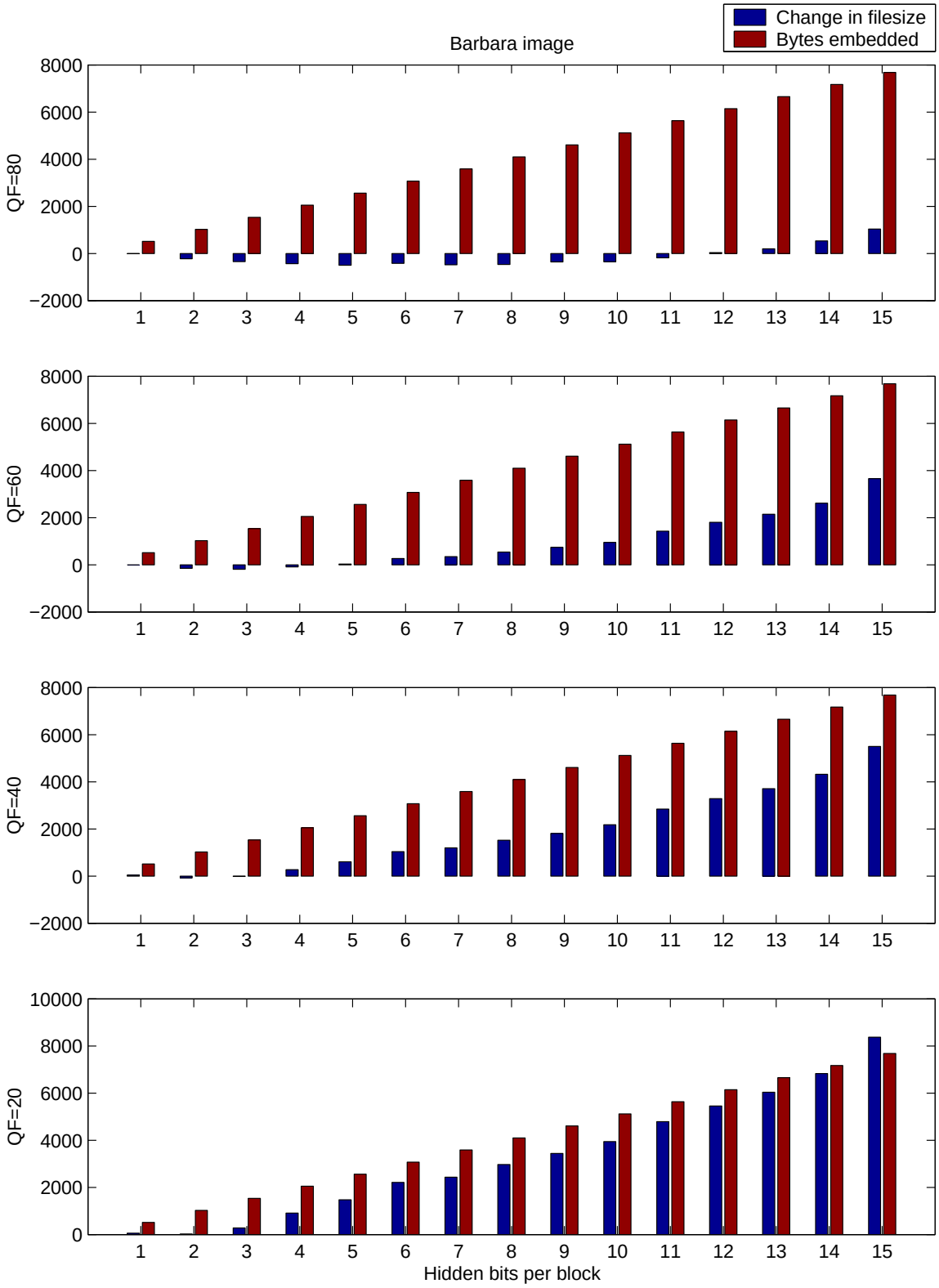


Figure 33: Filesize change for Barbara Image. The bitrates of the image from highest quality to lowest are {1.57, 1.08 , 0.8 , 0.54 } bpp.

Figure 34: Fishing Boat Image in 128x128, 256x256 and 512x512 dimensions

PSNR metric measures the quantization noise as intended. We list the compression rates and distortion values (before data hiding) in Table 8.

Table 8: The Compression Bitrate and the PSNR Values of Test Images

	128x128	**256x256**	**512x512**
Peppers			
JPEG QF	82	60	37
Bitrate (bpp)	2.35	0.98	0.49
PSNR (dB)	35.28	35.23	35.25
Fishing Boat			
JPEG QF	76	50	25
Bitrate (bpp)	2.06	0.98	0.45
PSNR (dB)	32.69	32.61	32.70
Lena			
JPEG QF	77	60	19
Bitrate (bpp)	2.15	1.05	0.35
PSNR (dB)	32.72	32.75	32.76

At the next step, randomly generated data is embedded into each image. The reduction in the PSNR value due to embedding is noted. The results of this experiment is given in Table 9. The Fishing Boat images with 15 bits per block embedding rate is shown in Figure 35. From these results, we can observe that as the image dimension increases, the embedding process gets more challenging. This effect is expected, since an 8 fold increase in image dimensions is equivalent to mapping of the pixels to the 8x8 blocks. The JPEG algorithm works on 8x8 blocks irrespective of the image dimension. Smooth blocks is more difficult to hide data at. On the other hand 8 fold increase in dimensions results in 64 fold increase in the number of blocks. If the data hiding performance is not reduced by the same ratio, the data hiding remains as a viable option. In the next section, we examine higher resolution images to see this effect.

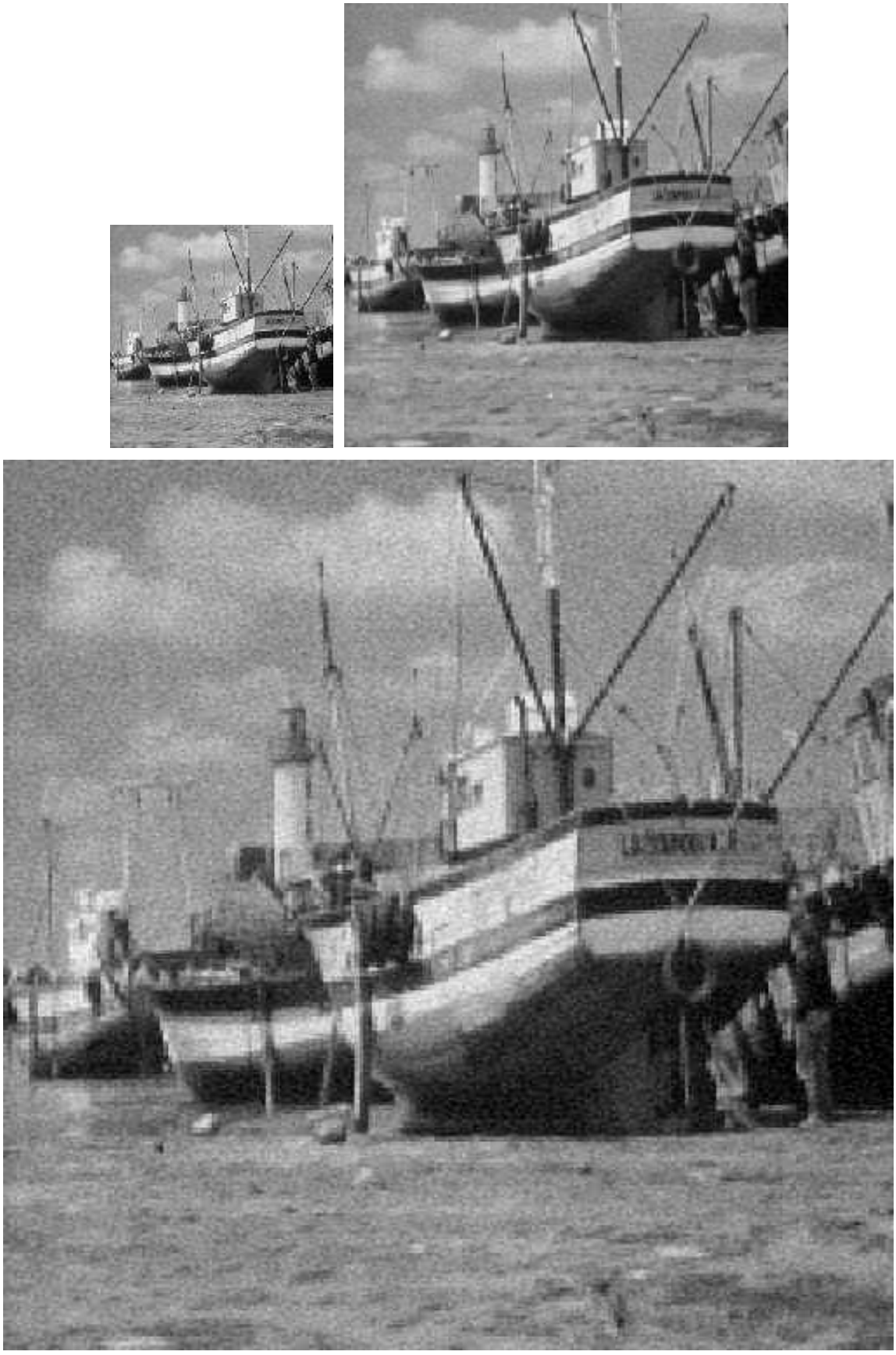


Figure 35: Fishing Boat Image with 15 hidden bits per block at the resolutions of 128x128, 256x256 and 512x512

Table 9: PSNR Loss at Different Embedding Rates

Test Image	Embedding Rate										
	5	6	7	8	9	10	11	12	13	14	15
Peppers											
128x128	0.05	0.07	0.09	0.11	0.14	0.17	0.2	0.24	0.25	0.31	0.34
256x256	0.26	0.35	0.46	0.56	0.69	0.82	0.99	1.13	1.3	1.47	1.67
512x512	0.88	1.2	1.48	1.76	2.08	2.4	2.75	3.09	3.43	3.82	4.32
Fishing Boat											
128x128	0.05	0.06	0.09	0.1	0.13	0.16	0.19	0.23	0.27	0.31	0.35
256x256	0.22	0.3	0.4	0.5	0.6	0.72	0.84	0.98	1.13	1.28	1.48
512x512	1.09	1.45	1.79	2.07	2.43	2.8	3.18	3.54	3.92	4.34	4.84
Lena											
128x128	0.05	0.06	0.08	0.1	0.12	0.15	0.18	0.2	0.23	0.28	0.32
256x256	0.15	0.2	0.26	0.33	0.4	0.48	0.57	0.68	0.75	0.86	1
512x512	1.96	2.52	3.04	3.48	3.92	4.37	4.84	5.35	5.85	6.33	6.93

3.6.4 Data Hiding For High Resolution Images

The latest image processing applications such as the digital cameras or the high definition television use very high image resolutions. As the restrictions due to the bandwidth and the storage are removed, a shift towards the high quality image applications is expected. In this section, we present some results on the performance of the described method for high resolution images.

We have used two different images for this experiment. The first image is captured from a HDTV broadcast of a popular TV show (Jay Leno Show, NBC). The second image is a picture taken with a 3 Mega pixel digital camera. These two images can be seen in Figure 36 (The original color images are converted to the gray scale format for this experiment).

The JPEG compression is built on the signal packing capabilities of the DCT. We know that constant signals can be coded to a single coefficient at their DCT representation. The DCT is selected for the JPEG standard since it can not only compress

Angelina Jolie on Tonight Show

Soccer Match

Figure 36: High Resolution Images for the Hiding Experiment. The first image is the screen capture of a HDTV broadcast at the resolution of 1080x1920. The second image is a digital camera shot at the resolution of 664x816.

DC signals into a single coefficient, but it is also efficient at coding slowly varying signals around a DC level. In other words, the columns of the DCT matrix approximate the eigenvectors of the auto-correlation matrix of AR(1) sequences with high correlation values. Therefore if the neighboring pixels in a block are strongly correlated with each other (no edges), the DCT operation provides an efficient approximation to the statistically optimum KL decomposition, [28, Chapter 5.12]. We would like to note the dependency of the JPEG coding efficiency on the block smoothness in here. Similarly the block smoothness depends on the resolution of the input. As the resolution of an image is increased, the information content per block is decreased and the smoothness (redundancy) in a block is increased improving the JPEG's performance in bit per block sense.

The HDTV and the latest digital cameras operate on high resolution images. These two applications areas are aimed to produce a very high quality without any visible artifacts. It is important to note that if there is a visible compression artifact at any sub-region of the image, the high resolution value of the image diminishes. The flawless image reproduction is the goal of high resolution image applications (especially for the digital cameras). We have noticed that the JPEG compression at the quality factor levels below 80% causes a visible degradation at the blocks with the flying hair. Due to the high quality standards, the benefit of the compression can not be fully used at these applications.

The Table 10 shows the bitrate and the PSNR of the test images at different quality factor levels. The high definition screen capture has the resolution of 1080x1920. The digital camera picture has the resolution of 664x816. At a typical screen display resolution of 512 pixels in width (half-screen width of a popular computer resolution of 762x1024), the compressed images and the original image are indistinguishable from each other. When the images are displayed at their full resolution, the compression artifacts on the objects become noticeable below the quality factor of 80. We have

Table 10: Bitrate and PSNR of High Resolution Images at Different Compression Levels

	Jolie		Soccer	
QF	bpp	psnr	bpp	psnr
90	1.159	51.547	2.827	45.916
80	0.865	48.697	1.851	34.367
70	0.788	47.257	1.581	32.817
60	0.724	42.586	1.266	31.066
50	0.632	39.279	1.159	30.126
40	0.463	38.543	0.985	29.187
30	0.426	37.352	0.830	28.134
20	0.328	35.221	0.628	26.786
10	0.230	31.599	0.388	24.866

selected to work at 80% quality factor level for both pictures. We think that the practical compression rate for the HDTV pictures can be lower than 80% when the temporal dimension (masking in time) is taken into account. We would like to note that HDTV standard has a second resolution level with 16:9 aspect ratio which is 720x1280. As discussed before the critical compression rate depends two resolutions, the image and the display resolution. We do not expect to have the same compression rate (QF) at both HDTV resolutions.

In the Table 11, we examine the change in file length after the hiding operation. For both images, the induced distortion is totally imperceptible for 6 bits/block hiding. And it is very difficult to detect for 8 bits/block or 1 byte/block hiding. The file size for Jolie and Soccer images before data hiding (at QF=80) are 223981 and 129863 bytes respectively.

The distortion on the soccer picture is not perceptible to 1 byte/block. In other words, it is possible to interchange 6.5% ($8466/(129863+649)$, the amount of embedded data over the file size after hiding) of the image bits with data bits without any degradation in perceived quality or any significant change in file size.

For the HDTV image capture, the zero-distortion capacity can be as high as 1

byte/block. It seems to be possible to embed large quantities of data into the HDTV images (at least into I-Frames). We note that the compression efficiency for Jolie image is much higher than the Soccer image. We can explain this by the different texture of two images. Jolie picture contains many objects with smooth surfaces such as the furniture, floor and the screen at the background. These objects can be coded very efficiently. The observed relative increase in file size (in comparison with the soccer image) is not surprising, since the compression is more efficient at this picture. We note that that the increase in file size is still less than the data embedded at all cases.

Table 11: Change in File Length After Embedding

Hiding Rate (byte/block)	Embedded Bytes	Change in Filesize	% Change in Filesize
	Jolie		
4/8	16200	7733	3
5/8	20250	9380	4
6/8	24300	13000	6
7/8	28350	15118	7
1	32400	17697	8
	Soccer		
4/8	4233	225	0.2
5/8	5291	345	0.3
6/8	6350	397	0.3
7/8	7408	473	0.4
1	8466	649	0.5

3.6.5 Examination of the Delivery Priority for Content and Hidden Bits

In this section, we examine the change in the delivery priority of the hidden data with respect to the image data. In some rare, but important scenarios the delivery of the hidden data can carry more importance than the delivery of the content. One example of such scenarios is the case where the hidden data backlog exceeds the capacity the data buffer of the server. At this scenario, the server should try to reduce the backlog in order not to lose any incoming hidden data bits. Other scenarios can be the

delivery of the application information which is of critical nature as in the content management application.

The hiding algorithm presented in this thesis operates at *any* embedding and compression rate pair. To transmit more hidden information, one only needs to instruct the algorithm to increase the number of bits embedded per block. A trade-off comes into the play, as we embed more bits into the image. Embedding more bits into the carrier image eventually causes a significant increase at the length of the composite image. This change can be around 40% at high embedding rates. We present the change in the file length due to the hiding operation for the peppers image in Figure 37. In this figure, peppers image is compressed at 1 bpp (QF=60) and the file length before hiding is 8105 Bytes. Assuming that 5% change in file length (after hiding) is a permissible number, we can conclude from this graph that we can embed up to 8 bits per block to the pepper image. As can be seen from the same graph, the change in the embedding rate significantly increases the file length for higher rates. Embedding 8 bits per block extends the file size to almost 8500 bits (1.04 bpp). The 88% of these bits is used for the content and the %12 is reserved for the hidden data.

To deliver higher bitrates of hidden information without excessive bandwidth usage, the transmitter may prefer to reduce the bits allocated to the content. For this example, if the transmitter reduces the compression rate to 0.75 bpp (QF=40) the file size before hiding operation becomes 6287 Bytes (file length of the image with no hidden information). If the transmission bitrate ceiling after hiding is set to 8500 bytes as before, a simple calculation shows us that the file size after hiding can be changed up to 35%. The second part of the Figure 37 shows that it is possible to embed 18 bits per block under these conditions. With 18 bits per block embedding, the bitrate of the composite image is 1.04 bpp. The 73% of these bits are allocated to image and the rest is used for data communication.

The two composite images at the same overall bitrate but having different image/data bits ratio is shown in Figure 38. The process described in here is analogous to the source-channel coding for multi-media signals. At source-channel coding, the number of the error correction bits can be increased or decreased depending on the noise level. Similarly depending on the significance of the hidden data, we can change the ratio of content/hidden bits. Since the proposed algorithm is flexible in hiding and compression rates, it is a straightforward task to change this ratio.

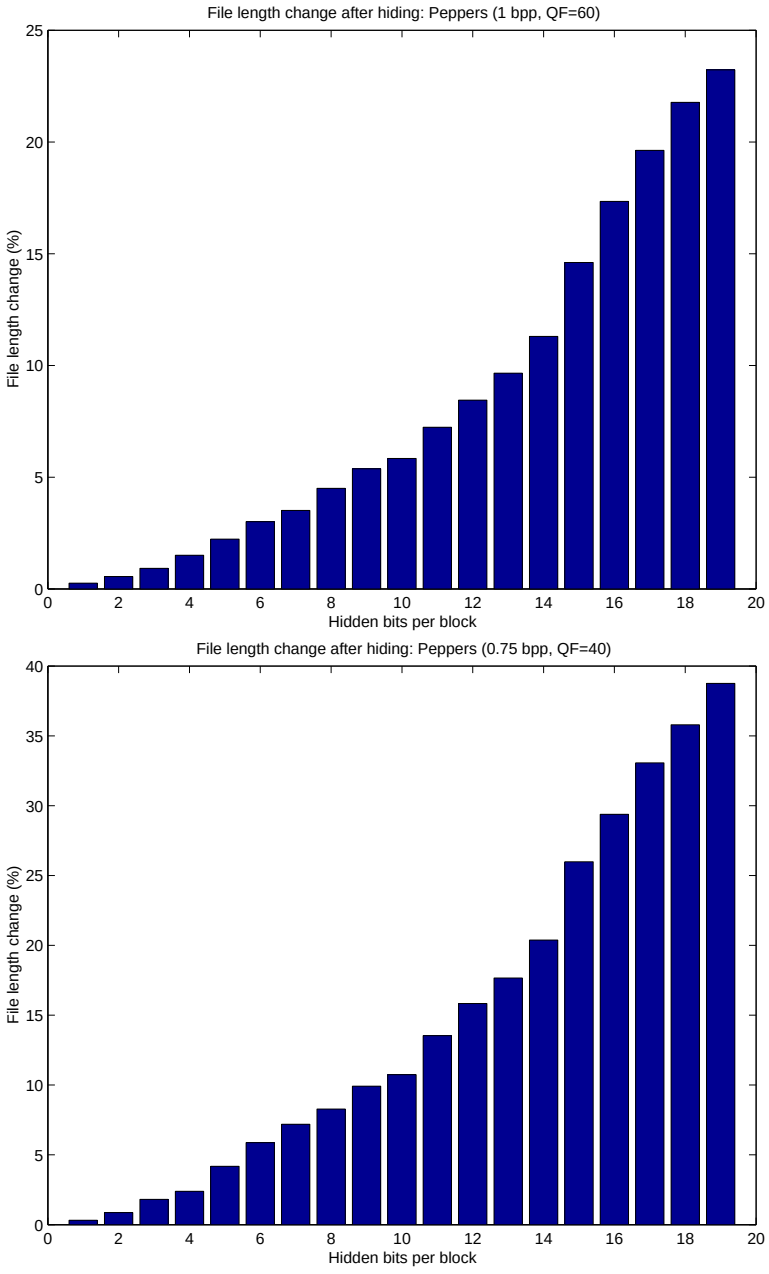


Figure 37: Change in Filesize for Peppers Image Encoded at 1 and 0.75 bpp

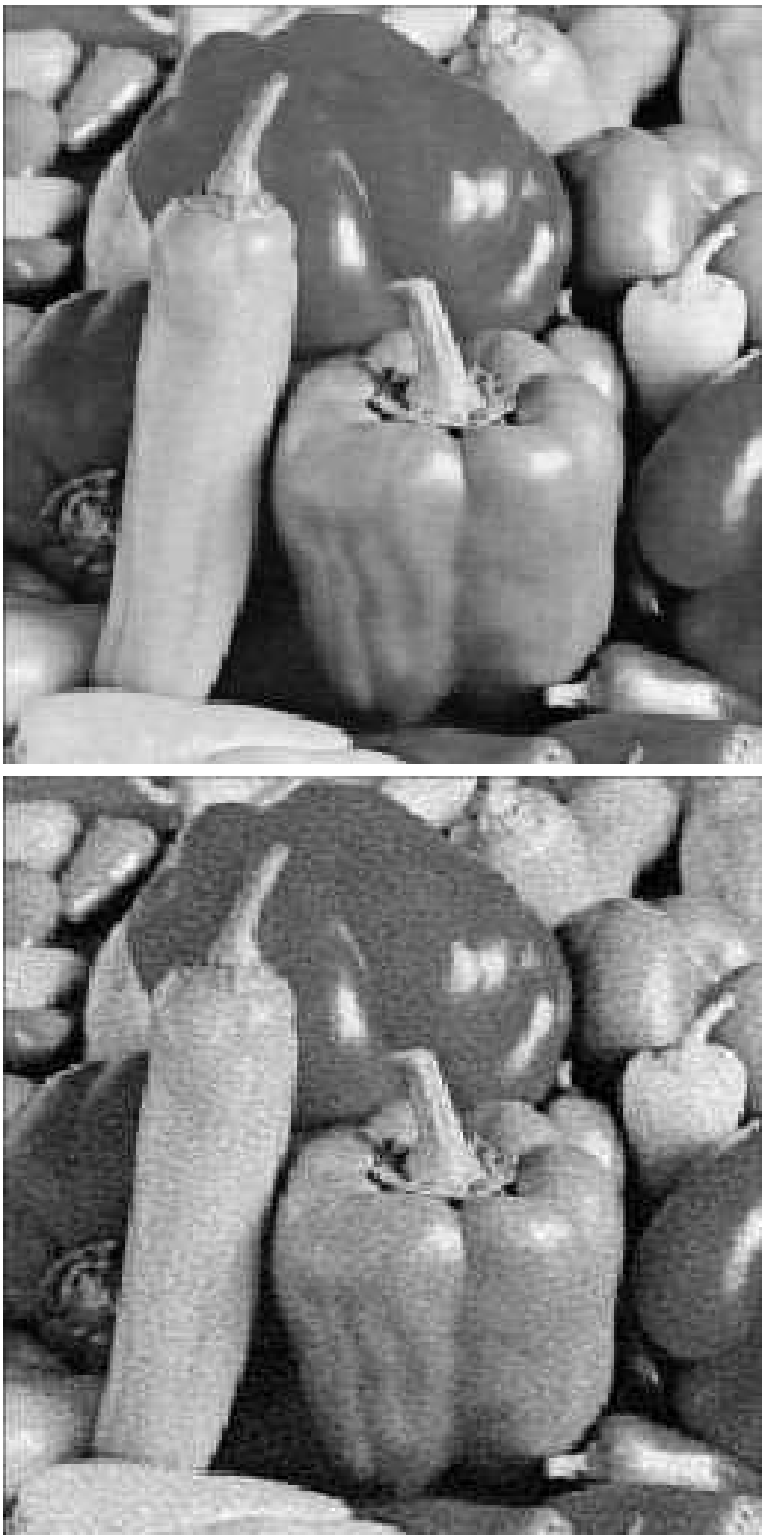


Figure 38: Two composite images with the bitrate of 1 bpp. The 12% of the total number bits of the top image is allocated to the hidden data. The number for the second image is 27%.

CHAPTER IV

AN IMAGE AUTHENTICATION APPLICATION

Multimedia security is an emerging research area. Message authentication and identity verification is an important application sub-group of the multimedia security. In this chapter we present a data hiding based image authentication method, [11]. The roots of the need for multimedia security applications can be associated with the existing computer network infrastructure which has been built without the guiding of a central authority, on a self-sufficient autonomous basis which is a lot easier to maintain and enlarge. The computer networks in use that is communication protocols operate on the assumption that parties on the network are truthful and willing to cooperate with each other to forward messages from one end of network to another. With the expansion of internet, the security of messages becomes a concern for many. The identity theft, reproduction and misuse of genuine messages, production of counterfeit messages is worrisome to all of us who uses internet for as one of main communication methods.

In this chapter we describe a data hiding based authentication algorithm for JPEG images. The algorithm is based on the embedding method described in this thesis. The application therefore inherits the minimum distortion embedding feature. It also has the additional security and tampering localization features which are the features specific to the this application sub-group.

The authentication method is based on two layers of authentication providing the features of security and tampering localization separately. The proposed method

modifies an input JPEG file to another valid JPEG file to embed authentication checkbits. The modification on the JPEG image is done by a minimally distortive data hiding algorithm.

The security aspect of the algorithm is provided by chaining 128 blocks of image together by the MD5 hash function. The hash function output is embedded into the blocks that the hash is generated from. In this way, an attack modifying coefficients of a single block causes authenticity problems along the whole chain. Chaining large number of blocks with MD5 function discourages attackers and therefore establishes the security leg of the system.

The localization feature is provided by a chain of length two. Due to the short length of the chain, this layer has almost no security. But the goal of this layer is provide attack localization with a short chain. Attack localization information is provide on a block basis by cyclic-redundancy-check (CRC) polynomials.

Finally, we introduce the user keys in order to implement the sender identity verification feature. User keys has been integrated to the system without user keys with a simple pre-processing stage.

The critical aspect of the application is the embedding algorithm. The embedding algorithm used presents multiple options to embed a given set hidden bits (authenticity checkbits). The embedding algorithm without constraints seeks to find the embedding option with the least distortion cost. As described later in the chapter, the multiplicity of options has turned out to be very useful at this application. At some occasions, we have deviated from the least distortive option and used the second least distortive option in order to provide authenticity.

The proposed authentication system is designed on top of JPEG compression system and fully compatible with the existing JPEG compressor/decompressors.

4.1 Authentication Algorithm

The proposed system uses a data hiding technique to embed checkbits into a JPEG file. The authenticated JPEG file is transmitted to the receiver. The receiver decodes the received JPEG image, checks its authenticity and detects the tampered blocks of the image. The system diagram of the proposed authentication-compression system is given in Figure 39.

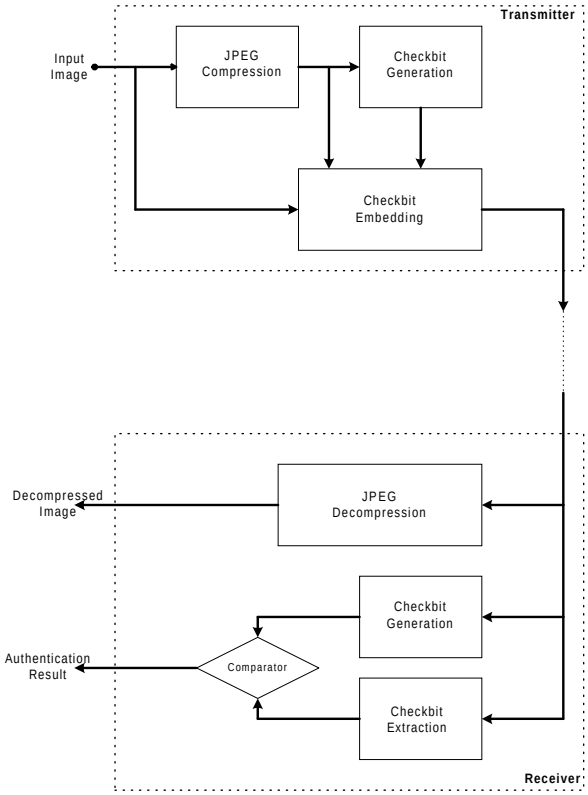


Figure 39: Proposed Compression-Authentication Framework

The image authentication systems can be categorized into three branches. The first distinguishing feature is the operation domain. The initial image authentication systems are designed to operate at the pixel domain [53]. The shortcomings of the first techniques have been analyzed [26] and some improved ones have been proposed [12]. The pixel domain authentication techniques cannot be used at communication applications. Such techniques are more suitable for storage applications, or for some

other applications requiring high image quality, like the trustworthy digital camera application of [23].

The second distinguishing characteristic is the method of inclusion of the authenticity checkbits into the image. The checkbits can be explicitly stored at the header of an image file or can be chosen to be embedded to the image by an appropriate data hiding technique. The drawback of data hiding is that due to the embedding operation, the image coefficients are perturbed from their original values to accommodate the hidden bits. This perturbation causes degradation of image quality. The drawback of appending techniques is that appending extends the length of the file by the number of checkbits. An additional hazard to appending may occur if transmitters decide to discard the header information (i.e. appended information) to save bandwidth. We think that the ubiquitous delivery of checkbits and their guaranteed co-existence with the image data is a desired or at least a convenient feature for practical image authentication systems.

The third distinguishing characteristic is the tampering localizaton capability. Some authentication systems provide a global answer to the image authenticity question. Others provide information about the tampering location. The detection of tampering location is a feature specific to the image authentication, non-existing in text messages.

We present a taxonomy of authentication systems known to us according to this classification. The pixel domain authentication techniques shown in Figure 40 operate on blocks to detect the location of tampering. Wavelet based techniques make use of multiresolution structure for the same purpose. The method proposed in this paper falls into the category of JPEG domain authentication methods through data embedding with the tampering location capability.

A technique similar to ours have been proposed by Wu and Liu [56]. The method of Wu modifies the quantized DCT coefficients of the image to embed checkbits.

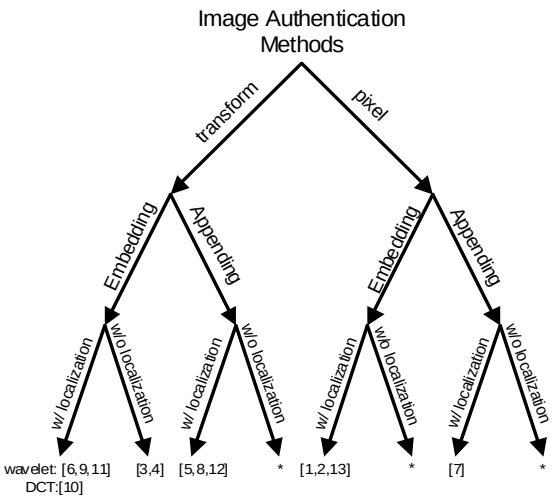


Figure 40: Authentication Algorithms in the Literature

Two checkbits are embedded per block. One checkbit is derived from the block and the other one is assigned by the sender. The derived and assigned checkbits are repeatedly embedded into each block to increase the robustness of the hidden bit extraction process. A shortcoming of this method is its insufficient security against the replacement attacks (the vector quantization attack as described in [26]). A simple replacement attack for this system is exchanging the positions of two JPEG blocks with the same hidden information. Since the number of checkbits per block is very few, the security of the algorithm can be compromised.

To deter replacement attacks without using an excessive number of checkbits, we propose a system with two layers of protection. According to our knowledge, the problem of joint security and tampering detection capability has not been attacked from this direction prior to us. The first layer generates a set of checkbits from a large number of blocks chained together. The embedded checkbits are used to provide security to the whole chain which they are derived from. A modification on anyone of the combined blocks is highly likely to violate the authenticity of the whole chain. The drawback of this layer is its low tampering detection capability (due to chaining). The second layer provides a low security authentication technique with

high tampering detection capability. The work of Wu and Liu [56] resembles the second authentication layer of our method. The combination of two layers is succeed by the flexibility of the data hiding algorithm that we have employed.

The challenge of the project is to accommodate two separate layers without destroying the authenticity of anyone of them. An additional change exists, since the compressed image are severely limited in the number of checkbits that can be embedded without causing an excessive distortion. To resolve the distortion considerations to some effect, we use the minimally distortive data hiding technique proposed in [9]. This technique blindly embeds a given set of hidden bits to a JPEG compressed image in a minimally distortive fashion. Different from other methods in the literature, the method presents multiple embedding options to hide a single bit. The encoder selects the option with the minimum embedding distortion. The decoder can extract the hidden information without the knowledge of option values. We have modified the described embedding technique so that both layers of authentication can be accommodated without any conflicts.

The next section describes two authentication layers in detail. The following section presents a simulation to illustrate the idea and then introduces the user keys to the authentication problem in a security improving, system supporting manner.

4.1.1 First Layer

The security of a text message is provided by chaining many message components together. After chaining, a modification on a single component causes the authenticity problems along the whole chain. Hash functions have been designed in cryptography [46] to chain sub-components. Hash functions or message digest functions are many-to-one functions at which a minor modification of the input causes a significant difference at output. For a cryptograhyically secure hash function, it should be computationally infeasible to find a message with a desired hash.

One of the most successful hash functions is the MD5 function, [42]. At this paper we use the implementation given at [42] with the 128 bit hash.

Similarly to the text hashing, we chain 128 JPEG blocks (8×8 blocks) together by applying MD5 function. The resulting 128-bit hash output is embedded into the same JPEG blocks (that the hash is generated from) at 1 bit per block rate. As shown in Figure 39, the decoder does the authenticity check by comparing the hidden hash bits with the calculated ones from the received and possibly tampered image.

It should be noted from Figure 39 that the hash at the transmitting side is calculated from the original JPEG blocks, but the hash at the receiving side is calculated from the modified JPEG blocks after authentication. For a sensible authenticity check, the hash value calculated at the transmitter and receiver should have the same value.

The traditional approach to overcome this consistency problem is to hash all of the components except 128 bits of the input (deallocation of 128 bits from hash input). The unhashed bit locations (known globally) are used to store the generated hash output. The system proposed by Wong [53] has the hash length of 64. Wong proposes to null the least significant bits (LSB) of every pixel of an 8×8 block before the hash calculation. The hash is calculated after the nulling operation and it is inserted to the LSB of every pixel in the block.

We do the deallocation in an indirect way. To establish the deallocation, an image feature is determined and checkbits are embedded in such a way that the feature remains unchanged after embedding. The feature, invariant to the embedding operation, can be considered as a footprint of the image that is to be authenticated. In the previously described scheme of Wong the first 7 bits of each pixel is the invariant feature.

The feature is calculated as follows:

- Classify the quantized DCT coefficients[1] of the JPEG image into two significance classes. A DCT coefficient is classified as significant if its quantization value has a magnitude higher than two times of its quantization step-size. In other words, the insignificant quantized coefficients are the ones that are quantized to $\{0, -1, 1\}$.

- Write the sign and location of the significant coefficients to a matrix. This matrix contains only three symbols. These symbols are $\{+SC, -SC, 0\}$ representing positive-, negative- and non-significant coefficients respectively.

The checkbits are derived from the significance matrix of 128 blocks. The hash of each run of 128 blocks is inserted back into the same blocks using the data hiding algorithm in [9]. We modify the algorithm in [9] to guarantee that all significant or non-significant coefficients remain at the same significance level after embedding. This resolves the conflict problem previously described. The flexibility of the hiding algorithm, i.e. possibility of having multiple options to embed a single bit, permits us to have this freedom at design.

In Figure 41, we present an illustration of the image footprint. The image on the left is Lena image JPEG compressed at the quality factor of 75. The image on the right is derived from significance matrix [2]. Our aim in giving this picture is to show that the feature we selected is a faithful representation of the general characteristics of the image. Similar features like the edge maps have been previously used in the literature [57].

Two types of attacks can be considered for the first layer: In case of non-intentional

[1]The JPEG compression system applies different quantization step-sizes for each DCT coefficient. The multiplicity of quantization step-sizes for each DCT coefficient is stored in the JPEG files. By the word quantized DCT coefficients, we mean the multiplicity of the quantization step-size.

[2]To derive this image, we have created a matrix whose $+SC$ coefficients are represented with 1, $-SC$ are represented with -1 and the non-significant coefficients are represented by 0. The image on the right is the inverse block-wise DCT of this image followed by two level histogram equalization (for high contrast).

Figure 41: Left: Lena Image compressed at quality factor of 75. Right: The hash function input in space domain.

attacks such as additive noise, the significance matrix described is not likely to be changed. Therefore calculated hash after a non-intentional attack and embedded hash should be the same. In case of a deliberate attack such as a replacement attack, it is expected that the block under attack have coefficients significantly different from the ones before the attack. Any tampering on the significance level of coefficients causes many mismatches between hidden and calculated hash values at the receiver letting us catch the misrepresented pictures [3].

4.1.2 Second Layer

The second layer of authentication is intended to detect the precise location of tampering. Without the second layer, any tampering on a single block causes authenticity problems along the chain of 128 blocks. The second layer is aimed to reduce this uncertainty.

At the second layer, we can not chain many blocks as done in the first layer. The most precise detection approach can be authenticating each block independently.

[3]To keep the paper exposition as simple as possible, we are not introducing the concept of keys at this stage.

Even though this is a valid option, we prefer to authenticate two blocks at a time (chain of length two) to further deter the replacement attacks.

The number of checkbits per block is an important security parameter. It has been shown that hash functions with 64 output bits are prone to systematic attacks. For moderately compressed images, it is not even possible to hide 64 bits per block without causing a significant distortion on the block. Therefore it is very difficult to establish the security of a given block when the operation domain is compressed images. With the two layer approach, the security of the chain that a block belongs to is established by the first layer, therefore at the second layer it is possible embed far less bits needed for block security.

We have chosen to use 5 checkbits per block at the second layer. As the number of checkbits suggests, there are only 32 possible checkbit combinations and irrespective of the hash method used, it is computationally trivial to find a counterfeit block with a given checkbit combination. But since the security is not the concern of this layer, we prefer to use error-detection-codes instead of hash functions to detect the tampering locality.

We use cyclic-redundancy-check (CRC) codes for checkbit generation of layer two. The CRC codes are used at many applications to detect the transmission error. We invite readers to [4] for more information on theory and applications of CRC codes.

CRC codes are described by a generator polynomial. The received word is a valid CRC codeword if and only if the generator polynomial is a factor of the received word polynomial. A suitable generator for the 5th degree CRC code is $g(D) = D^5 + D^3 + D + 1$. As suggested in [4, page 64], this polynomial is calculated by multiplying a 4th order primitive polynomial by $D + 1$ and has been used at other applications.

The systematic calculation of checkbits according to a generator polynomial takes

place as follows:

$$r(D) = \text{rem}(D^5 x(D), g(D))$$

In this equation $r(D)$ represents the checkbits and $x(D)$ represents the input polynomial[4].

One important performance criteria for a CRC polynomial is its error-detection coverage [51, page 124]. Assuming that a valid CRC codeword is transmitted at the encoder, the probability of channel noise moving the transmitted codeword to another valid codeword (case of missed error) is called error-detection coverage. Good codes are guaranteed to detect large number of systematic (like burst errors) and non-systematic errors. The overall error detection coverage of a CRC code is determined by its order. For the 5th order CRC polynomial, the error detection coverage is $1 - 2^{-5} = 0.9688$ implying that 96.8% of the moves from a codeword end at invalid codewords. We believe that this level of accuracy for tampering detection is sufficient for our purposes.

Until now, we have defined the checkbit generation method for the second layer. Now we define the feature for the second layer from which checkbits are generated. The following feature is derived from two consecutive blocks. The feature is based on the relative ordering of the DCT coefficients. The feature of the block is calculated as:

- Number the consecutive image blocks (row-wise)
- Start comparing the 32 lowest frequency DCT coefficients of the block.
- A vector of length 32 is formed by comparing the value of the quantized coefficients of the current block with the previous block. If a coefficient of the current block is larger than the corresponding coefficient at the previous block, the corresponding element of vector is set to 1. Otherwise, it is set to zero.

[4]The coefficients of a binary polynomial forms the codeword.

- The comparison vector of length 32 is written as a 31th order binary polynomial $(x(D))$

The corresponding CRC codeword is generated from the binary $x(D)$ polynomial as described before.

The generated CRC bits is embedded into the same block using the minimally distortive hiding method described. As done in the previous layer, a precaution has to be taken in order not to destroy the feature that the checkbits are based on. More specificaly, the relative order of the modified DCT coefficients stay the same before and after embedding.

At this layer, we set our goal as to trap a minor tampering on the blocks. If there is a tampering in a given block, the feature of the second layer *can* be perturbed and if it is perturbed, the 96.8% of such perturbations are detectable. In the next section, we incorporate the user keys (which are needed in practice) into the system in such a way that the fragility of the second layer feature is improved and therefore the tampering detection performance of the system is increased.

4.2 Computer Experiments

We present two computer simulations on the scheme described. The first scheme implements the two layer system as described in the previous section. The second simulation introduces secret keys.

4.2.1 Simulation One: System Without Secret Keys

We have followed the method described to authenticate 256×256 Lena image. Lena image is first JPEG compressed. The quantized DCT coefficients of the image are passed to checkbit generation system of the first layer. The generated significance map of the image is divided into chunks of size 128 blocks. The MD5 function is used to calculate the 128 bit hash of each chunk. The calculated bits are embedded at 1

bit per block rate to the chunks that they are generated from. At this experiment the sequence of 128 blocks is formed by four row blocks of Lena image.

At the second layer, the amplitude comparison result of the DCT coefficients in two consecutive blocks is digested to 5 checkbits. These checkbits are embedded into the image using the same embedding algorithm.

We illustrate the experiment in Figure 42. The upper left figure is the compressed image before authentication and the upper right image is the one after authentication. The PSNR values before and after authentication are 34.53 dB and 34.35 dB respectively. The distortion is not perceptible.

We test the system with a replacement attack. The attack consists of interchanging two blocks of the Lena image. The bottom left image of Figure 42 shows the authentication result of layer one. The white blocks of this figure illustrate a mismatch at the calculated hash bit and the extracted hash bit from that block. It is clearly observed from this picture that the first layer of authentication has detected an attack at the top and bottom parts of the picture.

The bottom right image shows the authentication result of the second layer. The two white blocks on the top and bottom rows show a mismatch at second layer of authentication. The actual attack we initiated is the interchange of the block on the left of the two on the top, with the left one of the two on the bottom. Because we have chained two blocks in second layer, a pair of blocks are detected to be questionable. It is known for sure that first link of the chain is always tampered. The second link may or may not be tampered. The reason of chaining two blocks is made clearer by the next simulation.

4.2.2 Simulation Two: System with Secret Keys

Security through cryptography is provided by the system and secret keys. In practice key selection and management is as important as the system design. The traditional

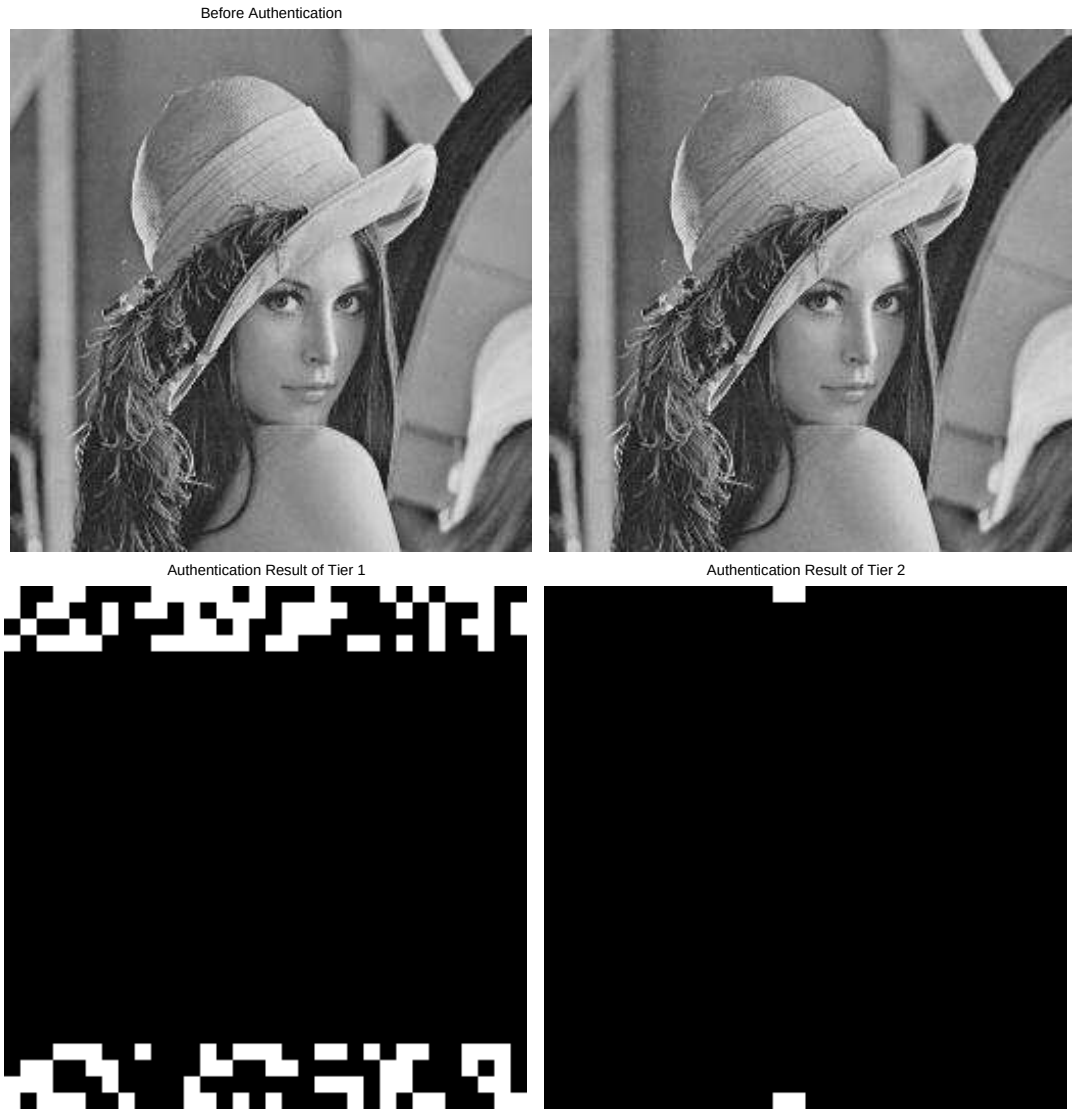


Figure 42: Upper Left: Lena image compressed at the quality factor of 75. Upper Right: Image after authentication. Lower Left: Authentication result of first layer. Lower Right: Authentication result of second layer.

way of authenticating text messages involves three components: The cover text, the hash output and the user-dependent keys to encode the hash. The encoded hash is appended to the cover text to facilitate the authenticity check. Keys can also be used to authenticate the sender's identity. Readers may refer to public-key encryption texts for details, [46].

The secret keys can be introduced in many ways. In order not to complicate the overall system, we present a very simple modification on the previous simulation. The keys are introduced by defining a key dependent shuffling stage which permutes the JPEG tiles of the image according to a randomly generated pattern depending on the secret key and possibly on some robust image feature like the one layer one. After shuffling picture of Lena image can be seen in Figure 43. After the shuffling stage, the algorithm described at the previous simulation is used as it is. Effectively, the shuffling stage randomizes the block scan order in a user dependent way.

Figure 43 shows the simulation result on the same image under the same attack. The PSNR of the authenticated Lena image is 34.36 dB. The tampering result of the first layer looks like a random looking pattern. Different from Figure 42, we can not easily say that the top and bottom parts of the image are questionable. Instead we can give two lists of 128 blocks which can be modified by the attackers. The second layer pinpoints the attack locations as in the previous simulation.

One may rightfully argue against the usage of length two chains at the second layer. The reason we have selected not to authenticate blocks independently is as follows: Since the data embedding and extraction method is publicly known, the embedded checkbits of each block is known by the attackers. Attackers can foil the second layer of authentication by replacing a block with another block having the same set of hidden bits. By introducing a relation between two blocks, we can thwart such replacement attacks and thus increase the overall tampering detectability.

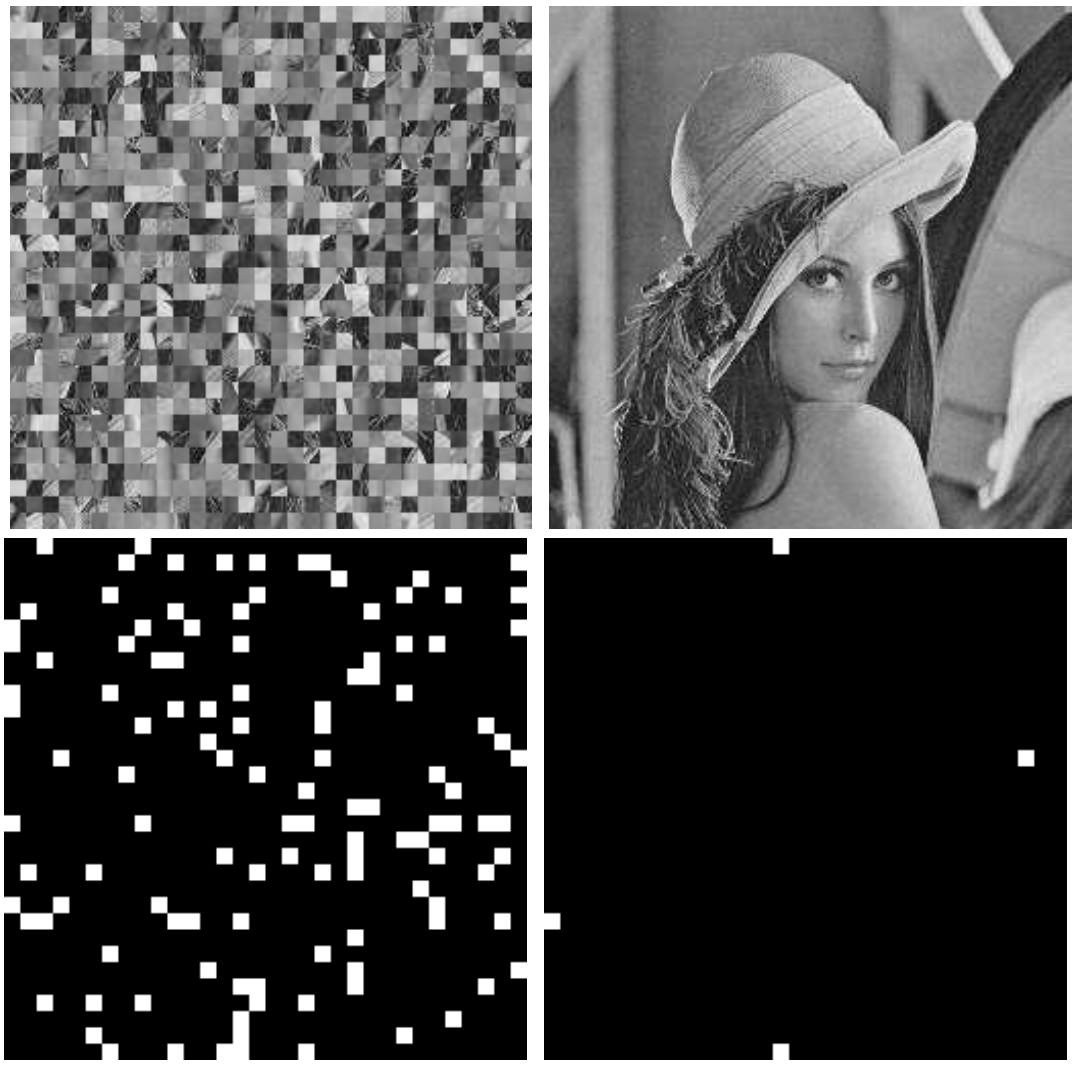


Figure 43: Upper Left: Result of Secret Key Dependent Randomization Stage. Upper Right: Authenticated Image. Lower Left: Authentication Result of Layer 1. Lower Right: Authentication Result of Layer 2.

4.2.3 Comparison with the Existing Methods

In this section, we would like to compare the proposed method with the existing methods in the literature. The word watermarking has been used for copyright protection application for quite a time. Data hiding is a more general term which includes copyright research and other areas of application. Authentication application falls in the area of watermarking. The watermarking methods can be separated into two. The robust watermarking methods are designed for copyright protection applications. The fragile watermarking methods are designed for authentication type applications. The testing and performance of evaluation of robust watermarking method has also been developed in time and a common set of attacks has been determined to compare different robust watermarking algorithms. The Stirmark program with tens of different attack options is a currently an accepted generic (algorithm independent) attack tool, [39]. The fragile watermarking systems can not be tested with the tools of robust watermarking algorithms. Fragile watermarking tools are built to detect the existence of an attack or tampering. Therefore these methods require significantly different metrics for comparison.

The first and by far the most important performance criteria for the authentication algorithms is the security. A secure authentication method should guarantee that a generation of an authentic looking forgery is extremely difficult. This aspect of performance evaluation (cryptographic security) is one of the lagging research subareas of watermarking research. The following is taken from a watermarking forum at which the future of watermarking is discussed, [21]. At this Forum Dr. Memon pointed out that "However, we are still far from being able to quantify or even analyze the security of current watermarking systems. It would be naive to expect provable security. But it should be noted that most commonly used cryptographic techniques only offer computational security. And even to achieve this, it takes years of analysis

and rigorous testing before one starts to gain some assurance. However, current watermarking literature seems to lack serious efforts at even defining the threat model and offering an appropriate analysis."

The other performance measures carry very little practical importance without the security aspect. The other features can be listed as tampering localization detection, type of tampering (filtering or malicious replacement), robustness to a set of operation (semi-fragile watermarking), computational complexity (for video applications). We list main authentication algorithms in Table 12 to compare the performance of the proposed algorithm.

Table 12: Comparison of the Authentication Algorithms in Literature

Method	Embedding	Localization	Compression Domain	Security
[58]	×	√	×	×
[59]	√	√	×	×
[53]	√	√	×	?
[12]	√	√	×	?
[34]	√	√	√ (Wavelet)	×
[31]	√	√	√ (Wavelet)	×
[20]	√	×	√	×
[22]	√	×	√	×
[33]	×	√	√	×
[5]	×	√	√	×
[57]	√	√	√(Wavelet)	?
[54]	×	√	√	?
[56]	√	√	√	×
Proposed	√	√	√	?

Many trade-offs can be observed from Table 12. It is clearly visible that the critical security feature is difficult to satisfy. Many algorithms in the literature have no security as soon as the workings of the algorithm is made public. The localization information is easier to provide for pixel domain or wavelet domain systems. DCT domain systems operate either in full-frame format, [20, 22, 33, 5], without any localization information or operate blockwise without any security, [56]. The proposed

algorithm operates claims to have both features. The proposed technique has two separate layers. The first layer establishes the security of the method by chaining 128 JPEG blocks together at the cost of having very little localization. The second layer establishes the localization information by chaining two blocks at the cost of having no security. By the suitable choices of embedded data (message digest) for both layers and the flexibility of the proposed embedding algorithm (multiple ways of embedding a single bit) we can accommodate both layers without any conflicts. To our knowledge, the proposed algorithm is the first one explicitly tackling the issues of security and localization for DCT based systems. The critical security aspect of the proposed method needs further scrutinity as done for [33] at [41]. In the authentication literature, the issue of authentication distortion for embedding systems has never been discussed until now. Our method also provides a provably distortion efficient way for authentication. The required embedding rate of the algorithm is 6 bits per block making the embedding imperceptible for moderately and lightly compressed images. The embedding rate can be increased to provide higher security if needed.

CHAPTER V

CONCLUSIONS

We have presented a new data hiding method for images. The method is designed to minimize the embedding distortion on the carrier image. To our knowledge, there is no other method in the literature explicitly designed for this purpose. Most hiding methods are designed to maximize the attack robustness for which the copyright protection is the main application area. An alternative way of expressing our goal is the maximization of embedding capacity at a given distortion tolerance.

On Robust Data Hiding: The data hiding research has initially focused on the robustness of the embedding technique to the deliberate or accidental attacks. Numerous methods have been proposed for this purpose. Unfortunately none of these methods have been found practical for the copyright enforcement applications. The studies have shown that it is relatively straightforward to embed a code in an audio or video file to counter a specific attack, but it is significantly more difficult to design an algorithm robust to many attacks. Some algorithms have found use at the non-hostile adversary applications for which the attack set is limited. An example of these applications is the recognition of the embedded codes from printed, scratched or not properly scanned images.

On the Project: The designed hiding system has three input parameters: Uncompressed image, compression rate and data to be hidden. The output of the system is a JPEG image with the embedded hidden data at the given compression rate (Quality Factor of the JPEG system). The system is scalable in embedding and compression rates. We should note that there are embedding algorithms in the literature robust to the JPEG compression. These algorithms are designed to preserve the hidden data

when the carrier is compressed. Although our method operates in JPEG domain, it differs from these algorithms in the sense that it does not aim to achieve robustness for further compression, but it aims to minimize the visual effect of hiding.

The encoder of the designed method embeds the given data into a JPEG file. The composite image is expected to travel through the communication channel. The receiver decodes the picture using the conventional JPEG decoder. The hidden data decoder extracts the hidden bits from the received JPEG file without any other side information.

In this project, we have first examined the viability of the JPEG domain hiding. We have presented an information theoretical derivation of a claim of hiding capacity saying that there exists room for data hiding if the images are not perfectly compressed. We have implemented some experiments on the conventional JPEG images to examine the relation between the hiding capacity and the embedding distortion. In these experiments, we have adopted the JPEG compression with Watson's human visual system model update as the perfect perceptual coder. We have seen that the left-over redundancy in the conventional JPEG images is sufficiently large at medium, high quality compression rates when compared with the perfect coder.

In order to minimize the embedding distortion, we have proposed a system with multiple degrees of freedom. These variables are the partitioning strategy, the distortion metrics and the human vision system model parameters. To find a good partitioning strategy, we have derived the expected distortion per hidden bit relation for a given partition. Next, we have conducted an exhaustive search to find the best strategy at different compression-hiding bitrates. Given the information about the optimum partitions at specific compression-hiding rates, we have proposed an ad-hoc partitioning rule for all compression rates. We have shown that this rule tracks the optimum partitioning strategy closely.

Distortion metrics is critical for the decision making at embedding. We try to

incorporate the existing knowledge of human visual system models into the design. The initial design uses MSE (PSNR) at embedding. This metric has been weighted with the Just Noticeable Difference levels derived from Watson's human visual system model. The weighted MSE has been tested by subjective experiments and the subjective tests favor its application.

The initial method embeds the same number of hidden bits at every 8 by 8 block. The local characteristics of an image varies a lot from block to block. We have developed a system for the input adaptive embedding. Some blocks with no apparent structure (noisy blocks) or blocks with low contrast are less prone to the embedding distortion. We have proposed a method to make better use of these blocks. With this method it is possible to embed more bits at the low contrast and high activity regions and less bits at the little activity, smooth regions of the image.

A significant challenge of the project is that the decoder should operate blindly that is without any side information. The multiple option embedding solution described in the thesis solves this problem and provides a search space for the distortion minimization. The options are generated through the partitioning operation. To hide the bits, the encoder searches the best approximation to the original signal in an option class selected by the hidden bits. It is important to populate each class in such a way that the distortion after embedding is minimized. In this document, we have shown that there are more than 1500 different options to hide three bits in a JPEG block. By fixing the partitioning strategy to a globally known strategy, we satisfy the requirement of blind operation while keeping the search possibility for minimum distortion embedding.

Analogies: The hidden data decoder can be interpreted as the syndrome decoder of an error-correction system. We have underlined the similarities between our method and the syndrome decoding. In the error-correction literature, the syndromes are the labels of the equivalence classes of the received input. The equivalence class of all-zero

syndrome contains the codewords of the error-correction code. The union of different syndrome sets form all possible combinations of bit sequences in an n dimension space. In this work, we focus on a partitioning mechanism to divide a 64 dimension space (8 by 8 block) in such a way that each partition (equivalence class) is fit to approximate the input well enough. The label of the class (syndrome) is the hidden information embedded. To underline the similarities between two disciplines, we have named an element of the partition as the set-leader (in resemblance with the coset-leader at error-correction coding). We have shown that the execution of the search over the set-leaders reduces the complexity of search significantly.

Another interesting analogy is the one between the information theoretical background of data hiding by Moulin and Chen [35, 14] and the decoding method proposed. The information-theoretic decoders operate on the basis of joint typicality. The concept of joint typicality is useful at proving asymptotical results, but it is not possible to implement a practical method based on the concept of typicality. In this work, the decoder of the proposed method essentially checks the class of incoming bits and this operation can be loosely interpreted as a typicality checker. Our work can be interpreted as a practical implementation of the theoretical work based on the typicality.

Subjective Tests: We have set up some subjective experiments to examine hiding capacity. The conclusion for Lena image is that there is no additional distortion perceived up to 5 bits per block embedding, when the compression bitrate is higher than 1 bpp. The same score for Baboon image is 7 bits per block. In other works, it is possible to embed almost one third of Declaration of Independence (whose total is 7922 bytes without signers' names) in 512 by 512 Lena image. The conducted experiments have shown that the high resolution images can carry significantly larger amounts of data even at low embedding densities. The performance of the method has been compared with the well known spread spectrum data hiding technique.

The experimental results have shown that the spread spectrum technique provides a competitive but less efficient performance.

Application Example: We have proposed an authentication application for JPEG images. The transmitting side embeds checkbits into the image before the transmission. These bits are used by the receiver to verify the integrity of the received content and the identity of the sender. The authentication method inherits the blindness and the minimum distortion embedding principle. In addition to these features, the authentication algorithm is designed for the cryptographic security and the tampering location detection.

Minimal Distortion Claim: We would like to emphasize that the word minimal in the minimal distortion title is being used in a restrictive sense. Under the necessary requirements of the communication applications (DCT domain embedding with blind decoding), the minimality of the distortion can be argued by the analysis given in Chapter 3. Similar designs can be developed for JPEG 2000 systems. We have focused on the JPEG compression because of its relation with the MPEG standards.

Extensions: The method can be easily extended to the MPEG systems. The MPEG standard uses JPEG compression at the one of its sub-components (I-frame coding). The proposed system can be directly incorporated to I-frame coding.

A more ambitious goal can be the joint design of compression and hiding operations. The joint design should have good compression performance when working at the compression mode and should have better performance than the separate design at the hiding mode. In here, we present a hiding method on JPEG compressed images. The hiding block is guided by the JPEG compression (quantization table, JND levels of DCT coefficients). For a doubly coordinated work, not only the guidance of compression on hiding, but also the guidance of hiding on compression is needed. For the scenarios at which the embedded data and the transmitted content have significantly different importance values (priorities) the need for doubly coordinated

arises. If the delivery of the hidden data is a lot more important than the delivery of the carrier, the hiding block becomes the guiding block of the compression. We have given an illustration of the spectrum of priorities in Section 3.6.5. The joint optimization of two blocks is another level above the doubly coordinated work. Joint work pursues a single block for compression-hiding. One may think this single block as a perfect image coder, which allocates the hidden bits and image bits by considering the relative importance of two sources and the significance level of deleted, transmitted and replaced image bits together. The research on perceptual coders for images is still ongoing. The joint hiding-compression block is linked to the perceptual coder research.

Remarks: The main application area of the proposed system is the data broadcasting. The high definition television broadcasting can be an interesting application area for the method. The embedding experiments on the high resolution pictures have shown that the hiding payload for the high definition pictures is significantly larger than traditional images in spite of a loss of embedding density. If the before- and after- hiding file length changes negligibly, the only cost of hidden data transmission is the embedding distortion whose visibility is minimized in this project. We hope that the presented work would find interesting applications in the future.

Contributions:

- An information theoretic proof of the hiding capacity claim as the difference between imperfectly and perfectly compressed signals is given, [8].
- A minimum distortion data hiding method for JPEG images satisfying the requirements of communication applications have been designed, optimized and tested, [10].
- An authentication application for JPEG images to upgrade the security of the

conventional JPEG system is proposed, [11].

- A multiple description coding scheme based on the presented hiding ideas have been designed, [7].

APPENDIX A

BEST PARTITIONING STRATEGIES

```
For JPEG Compression Quality Factor = 10

Quantization_Matrix =

    80    55    50    80   120   200   255   255
    60    60    70    95   130   255   255   255
    70    65    80   120   200   255   255   255
    70    85   110   145   255   255   255   255
    90   110   185   255   255   255   255   255
   120   175   255   255   255   255   255   255
   245   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255

 Embedding Rate : Distortion Value : Partition
 1 : Distortion = 32.63 :
     Partition = { [50  55  60  60  65  70  70  70  80  80 80  85  90  95] }
 2 : Distortion = 35.66 :
     Partition = { [50  65  70  70  80], [55  60  60  70  80  80  85  90  95] }
 3 : Distortion = 37.82 :
     Partition = { [50  80], [55   65   70   80   80  110  110],
                   [60  60  70  70  85  90  95] }
 4 : Distortion = 39.59 :
```

```
    Partition = { [50], [55  70  85], [60  65  80  80  90],
                  [60   70   70   80   95  110  110] }
 5 : Distortion = 41.21 :
    Partition = { [50], [55], [60  70  85  95], [60  80  80  80],
                  [65   70   70   90  110  110  120  120  120] }
 6 : Distortion = 42.72 :
    Partition = { [50], [55], [60  80], [60  80], [65   70   90   95  130],
                  [70   70   80   85  110  110  120  120  120] }
 7 : Distortion = 44.15 :
    Partition = { [50], [55], [60], [60], [65  80  85], [70   70   90  110],
                  [70   80   80   95  110  120  120  120  130] }
 8 : Distortion = 45.59 :
    Partition = { [50], [55], [60], [60], [65  85],
                  [70   80   90  130], [70   80   95  120],
                  [70   80  110  110  120  120] }
 9 : Distortion = 47.03 :
    Partition = { [50], [55], [60], [60], [65], [70  85], [70   95  110  110],
                  [70   90  120  120  145], [80   80   80  120  130] }
10 : Distortion = 48.46 :
    Partition = { [50], [55], [60], [60], [65], [70 95], [70  110], [70  110],
                  [80   80  120  120  120], [80   85   90  130  145] }


For JPEG Compression Quality Factor = 20


Quantization_Matrix =


    40    28    25    40    60   100   128   153
    30    30    35    48    65   145   150   138
```

```
   35    33    40    60   100   143   173   140
   35    43    55    73   128   218   200   155
   45    55    93   140   170   255   255   193
   60    88   138   160   203   255   255   230
  123   160   195   218   255   255   255   253
  180   230   238   245   255   250   255   248

Embedding Rate : Distortion Value : Partition
1 : Distortion = 16.40 :
    Partition = { [25  28  30  30  33  35  35  35  40  40  40  43  45  48] }
2 : Distortion = 17.91 :
    Partition = { [25  33  35  35  40  55  55],
                  [28  30  30  35  40  40  43  45  48] }
3 : Distortion = 18.99 :
    Partition = { [25  40  55  55], [28  33  35  35  45],
                  [30  30  35  40  40  43  48] }
4 : Distortion = 19.89 :
    Partition = { [25], [28  35  40], [30  33  40  40  48  55],
                  [30  35  35  43  45  55] }
5 : Distortion = 20.71 :
    Partition = { [25], [28  45], [30  35  43], [30  40  40  40  65],
                  [33  35  35  48  55  55  60  60  60] }
6 : Distortion = 21.46 :
    Partition = { [25], [28], [30  40], [30  40  65], [33  35  43  48],
                  [35  35  40  45  55  55  60  60  60] }
7 : Distortion = 22.17 :
    Partition = { [25], [28], [30], [30], [33  40  43], [35  35  45  55],
                  [35  40  40  48  55  60  60  60  65] }
8 : Distortion = 22.89 :
```

```
     Partition = { [25], [28], [30], [30], [33  43], [35  40  45],
                   [35  40  48  60  65], [35  40  55  55  60  60] }
 9 : Distortion = 23.61 :
     Partition = { [25], [28], [30], [30], [33], [35  43], [35  45  60  60],
                   [35 48  55  55  73], [40  40  40  60  65] }
10 : Distortion = 24.31 :
     Partition = { [25], [28], [30], [30], [33], [35], [35 48], [35  55],
                   [40  40  55  60  60], [40  43  45  60  65  73] }



For JPEG Compression Quality Factor = 40

Quantization_Matrix =

    20    14    13    20    30    50    64    76
    15    15    18    24    33    73    75    69
    18    16    20    30    50    71    86    70
    18    21    28    36    64   109   100    78
    23    28    46    70    85   136   129    96
    30    44    69    80   101   130   141   115
    61    80    98   109   129   151   150   126
    90   115   119   123   140   125   129   124

 Embedding Rate : Distortion Value : Partition
 1 : Distortion = 8.31 :
     Partition = { [13  14  15  15  16  18  18  18  20  20  20  21  23  24] }
 2 : Distortion = 9.07 :
     Partition = { [13  15  18  18  20], [14  15  16  18  20  20  21  23  24] }
```

```
 3 : Distortion = 9.61 :
     Partition = { [13  18  20], [14  16  18  20  28  28],
                   [15  15  18  20  21  23  24] }
 4 : Distortion = 10.05 :
     Partition = { [13], [14  18  20], [15  16  20  21  24],
                   [15  18  18  20  23  28  28] }
 5 : Distortion = 10.44 :
     Partition = { [13], [14], [15  18  21], [15  18  20],
                   [16  18  20  20  23  24  28  28  30  30  30] }
 6 : Distortion = 10.80 :
     Partition = { [13], [14], [15  21], [15  20], [16  20  20  23  33],
                   [18  18  18  24  28  28  30  30  30] }
 7 : Distortion = 11.15 :
     Partition = { [13], [14], [15], [15], [16  20], [18  18  21  24  30],
                   [18  20  20  23  28  28  30  30  33] }
 8 : Distortion = 11.51 :
     Partition = { [13], [14], [15], [15], [16], [18  20  21  33],
                   [18  20  24  28  28], [18  20  23  30  30  30] }
 9 : Distortion = 11.88 :
     Partition = { [13], [14], [15], [15], [16], [18  20], [18  21],
                   [18  23  28  28], [20  20  24  30  30  30  33  36] }
10 : Distortion = 12.25 :
     Partition = { [13], [14], [15], [15], [16], [18  24],[18  23],
                   [18  28], [20  20  30  33], [20  21  28  30  30  36] }


For JPEG Compression Quality Factor = 80


Quantization_Matrix =
```

```
    6     4     4     6    10    16    20    24
    5     5     6     8    10    23    24    22
    6     5     6    10    16    23    28    22
    6     7     9    12    20    35    32    25
    7     9    15    22    27    44    41    31
   10    14    22    26    32    42    45    37
   20    26    31    35    41    48    48    40
   29    37    38    39    45    40    41    40

Embedding Rate : Distortion Value : Partition
1 : Distortion = 2.56 :
    Partition = { [4  4  5  5  5  6  6  6  6  6  6  7  7] }
2 : Distortion = 2.81 :
    Partition = { [4  5  5  6  7  7], [4  5  6  6  6  6  6] }
3 : Distortion = 2.99 :
    Partition = { [4  6  6], [4  6  6  9  9], [5  5  5  6  6  7  7  8] }
4 : Distortion = 3.13 :
    Partition = { [4], [4], [5  5  6  7  7  8], [5  6  6  6  6  6  9  9] }
5 : Distortion = 3.28 :
    Partition = { [4], [4], [5  6  6  7], [5  6  6  7], [5  6  6  8  9  9] }
6 : Distortion = 3.42 :
    Partition = { [4], [4], [5  6], [5  6], [5  6  8],
                  [6  6  6  7  7  9  9 10 10 10 10] }
7 : Distortion = 3.54 :
    Partition = { [4], [4], [5  7], [5  8], [5  7], [6  6  6  9 10 10],
                  [6  6  6  9 10 10] }
8 : Distortion = 3.66 :
    Partition = { [4], [4], [5], [5], [5], [6  6  7 10], [6  6  7 10 10],
```

```
                    [6  6  8  9  9 10] }
 9 : Distortion = 3.79 :
     Partition = { [4], [4], [5], [5], [5], [6  6], [6  6], [6  7  8 10 10],
                    [6  7  9  9 10 10] }
10 : Distortion = 3.92 :
     Partition = { [4], [4], [5], [5], [5], [6  6], [6  7], [6  7], [6  8 10 10],
                    [6  9  9 10 10] }
```

REFERENCES

[1] Alturki, F., *Theory and Applications of Data Hiding in Still Images*. PhD thesis, School of Electrical Engineering, Georgia Institute of Technology, 2001.

[2] Barron, R. J., *Systematic Hybrid Analog/Digital Signal Coding*. PhD thesis, Department of EE, MIT, 2000.

[3] Barron, R., Chen, B., and Wornell, G., "The duality between information embedding and source coding with side information and its implications/applications," *IEEE Trans. Information Theory*, submitted 2000.

[4] Bertsekas, D. and Gallager, R., *Data networks*. Prentice Hall, 1992.

[5] Bhattacharjee, S. and Kutter, M., "Compression tolerant image authentication," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 435–439, 1998.

[6] Ç. Candan, "A survey of information hiding," *Qualification Exam Report, Georgia Institute of Technology*, 2000.

[7] Ç. Candan, "A multiple description coding scheme based on the chinese remainder theorem," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2002.

[8] Ç. Candan and Jayant, N., "A new interpretation of data hiding capacity," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2001.

[9] Ç. Candan and Jayant, N., "A minimum distortion data hiding technique for compressed images," *IEEE Multimedia Signal Processing Workshop*, 2002.

[10] Ç. Candan and Jayant, N., "A minimum distortion data hiding technique for compressed images," *International Workshop on Multimedia Signal Processing*, 2002.

[11] Ç. Candan and Jayant, N., "A data hiding based secure authentication technique for jpeg compressed images with tampering localization capability," *International Workshop on Digital Watermarking, published by Springer Verlang at Lecture Notes in Computer Science Series*, 2003.

[12] Celik, M., Sharma, G., Saber, E., and Tekalp, A., "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Process.*, pp. 585–595, 2002.

[13] Chen, B., *Design and Analysis of Digital Watermarking, Information Embedding and Data Hiding Systems*. PhD thesis, Department of EE, MIT, 2000.

[14] Chen, B. and Wornell, G., "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Information Theory*, vol. 47, pp. 1423–1443, 2001.

[15] Chou, J., Pradhan, S., and Ramchandran, K., "On the duality between distributed source coding and data hiding," *Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1503–1507, 1999.

[16] Costa, M. H. M., "Writing on dirty paper," *IEEE Trans. Information Theory*, pp. 439–441, 1983.

[17] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*. John Wiley and Sons, 1991.

[18] Cox, I. J., Kilian, J., Leighton, F. T., and Shamoon, T., "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, pp. 1673–1687, 1997.

[19] Cox, I. J., Killian, J., Leighton, F. T., and Shamoon, T., "Ieee signal processing society best paper award at image and multidimensional signal processing.".

[20] Du, R. and Fridrich, J., "Lossless authentication of MPEG-2 video," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 893–896, 2002.

[21] Forum, "What is the future for watermarking (part 1)," *IEEE Signal Processing Magazine*, vol. 20-5, pp. 55–59, 2003.

[22] Fridrich, J., Goljan, M., and Du, R., "Invertible authentication watermark for JPEG images," *International Conference on Information Technology*, pp. 223–227, 2001.

[23] Friedman, G., "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Consumer Electronics*, pp. 905–910, 1993.

[24] Gelfand, S. I. and Pinsker, M. S., "Coding for channel with random parameters," *Problems of Control andInformation Theory*, pp. 19–31, 1980.

[25] Heegard, C. and Gamal, A. E., "On the capacity of computer memory with defects," *IEEE Trans. Information Theory*, vol. 29, pp. 731–739, 1983.

[26] Holliman, M. and Memon, N., "Counterfeiting attacks on oblivious blockwise independent invisible watermarking schemes," *IEEE Trans. Image Process.*, pp. 432–441, 2000.

[27] Huang, S. J., "Opening the floodgates for digital information services," *White Paper, Scientific Atlanta Corporation*, 2002.

[28] Jain, A. K., *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[29] Jayant, N., Johnston, J., and Safranek, R., "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, pp. 1385–1422, 1993.

[30] Jayant, N. and Noll, P., *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[31] Kundur, D. and Hatzinakos, D., "Digital watermarking for telltale tamper proofing and authentication," *Proc. IEEE*, pp. 1167–1180, 1999.

[32] Larsen, R. J. and Marx, M. L., *An Introduction to Mathematical Statistics and its Applications*. Prentice Hall, 2001.

[33] Lin, C.-Y. and Chang, S.-F., "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits and Systems for Video Tech.*, pp. 153–168, 2001.

[34] Lu, C.-S. and Liao, H.-Y., "Multipurpose watermarking for image authentication and protection," *IEEE Trans. Image Process.*, pp. 1579–1592, 2001.

[35] Moulin, P. and J.A.O'Sullivan, "Information theoretic analysis of information hiding," *IEEE Trans. Information Theory*, submitted 2000.

[36] Neumann, J. V. and Morgenstern, O., *Theory of Games and Economic Behaviour*. Princeton Press, 1947.

[37] O'Ruanaidh, J. J. K. and Pun, T., "Rotation, scale and translation invariant digital image watermarking," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 536–539, 1997.

[38] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[39] Petitcolas, F., "Watermarking schemes evaluation," *IEEE Signal Processing Magazine*, pp. 58–64, 2000.

[40] Podilchuk, C. and Zeng, W., "Image-adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Comm.*, vol. 16, pp. 525–539, 1998.

[41] Radhakrishnan, R. and Memon, N., "On the security of the digest function in the sari image authentication system," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 12, pp. 1030–1033, 2002.

[42] Rivest, R., "The MD5 message-digest algorithm," http://theory.lcs.mit.edu/ rivest/rfc1321.txt.

[43] Servetto, S., Podilchuk, C., and Ramchandran, K., "Capacity issues in digital image watermarking," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, vol. 1, pp. 445–449, 1998.

[44] Shannon, C. E., "A mathematical theory of communication," *Bell Sys. Tech. Journal.*, vol. 27, pp. 379–423, 1948.

[45] Shapiro, J. M., "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, pp. 3445–3462, 1993.

[46] Stinson, D. R., *Cryptography : theory and practice*. CRC Press, 1995.

[47] Tao, B. and Dickinson, B., "Adaptive watermarking in the DCT domain," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2985–2988, 1997.

[48] Tu, C. and Tran, T. D., "Context based entropy coding of block transform coefficients for image compression," *IEEE Trans. Image Process.*, Nov 2002.

[49] Wang, Z., Bovik, A. C., and Lu, L., "Why is image quality assessment so difficult ?," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2002. Demo at: http://anchovy.ece.utexas.edu/ zwang/research/quality_index/demo_lena.html.

[50] Watson, A., "Perceptual optimization of DCT color quantization matrices," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, vol. 1, pp. 100–104, 1994.

[51] Wicker, S., *Error control systems for digital communication and storage*. Prentice Hall, 1995.

[52] Wolfgang, R., Podilchuk, C., and Delp, E., "Perceptual watermarks for digital images and video," *Proc. IEEE*, vol. 87, pp. 1108–1126, 1999.

[53] Wong, P. and Memon, N., "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Process.*, pp. 1593–1601, 2001.

[54] Wu, C., "On the design of content-based multimedia authentication systems," *IEEE Trans. on Multimedia*, pp. 385–393, 2002.

[55] Wu, M., *Multimedia Data Hiding*. PhD thesis, Department of EE, Princeton University, 2001.

[56] Wu, M. and Liu, B., "Watermarking for image authentication," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 437–441, 1998.

[57] Xie, L. and Arce, G., "A class of authentication digital watermarks for secure multimedia communication," *IEEE Trans. on Multimedia*, pp. 242–252, 2001.

[58] Xie, L., Arce, G., and Graveman, R., "Approximate image message authentication codes," *IEEE Trans. on Multimedia*, pp. 242–252, 2001.

[59] Yeung, M. and Mintzer, F., "An invisible watermarking technique for image verification," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 680–683, 1997.